UNIVERSITEIT LEIDEN

Abstract

Faculty of Science
Department of Computer Science

Advanced Data Analytics

Building robust prediction models for Activity Recognition and Energy Expenditure from raw accelerometers using Gated Recurrent Units and Long Short Term Memory Neural Networks.

by Jeremiah Nana Kwabena OKAI

Human Activity Recognition (HAR) and Energy expenditure (EE) are active field of research in machine learning. Due to the recent advancement in technology, more accelerometer sensor devices are used to collect information that can be used to train models for HAR and prediction of EE. Training models for HAR and EE is a very challenging task because multiple activities are being performed by different people at a different pace and in different environments. In this research, we studied a novel approach used in building robust and highly accurate deep neural network models for HAR and prediction of EE. We used Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) approaches for building these models. The datasets used to train these models were provided by the Leiden University Medical Center (LUMC). This study involved 35 participants performing 16 sedentary, ambulatory and lifestyle activities in a semi-structured environment. The results obtained from the models built are competitive with the state of the art approaches.

Acknowledgements

As the popular saying goes, *"everything which has a beginning, also has an end."* After an intensive year, I personally want to say a big thank you to everyone who contributed directly or indirectly to the success of this project but below personalities deserve special recognition.

First of all, I want to say a special thank you to both Claudio Sá and Stylianos Paraschiakos. No amount of words could express how grateful I was to have you two as my daily supervisors. It was a great pleasure studying under you guys and I really learnt a lot. Claudio, you taught me how to critically analyze problems; pay attention to small details and how to develop solutions in tackling such problems. You also taught me how to write concisely and communicate effectively which I am very grateful for. Stylianos, you also taught me how to exercise patience during difficult periods and shared a lot of insight in your field of expertise for the success of this project. No amount of words can express how thankful I am to have the opportunity to work with such great minds.

Secondly, I want to say a big thank you to my supervisor, Dr. Arno Knobbe, I am really grateful for giving me the opportunity to work with you. I want to thank you a lot for all the advice and constructive feedback you gave me for making this project successful.

Thirdly, I want to thank Dr. Wessel Kraaij and Ricardo Cachucho who also gave me the opportunity to work with them during my research year and also directing me on this path. Ricardo, I want to personally thank you a lot for introducing me to Dr. Arno Knobbe, Claudio Sà and Stylianos Paraschiakos to work with and also having a lot of trust and patience with me while working with you.

Last but not least, my appreciation goes to the entire Brace family of *Tema* for being there always and to my nuclear family. Not forgetting the Molecular Epidemiology department of Leiden University Medical Center who gave me the opportunity to work as an intern and also contributed to the success of this project. It has been a wonderful journey. God richly bless you for you all.

Jeremiah Nana Kwabena Okai

I want to dedicate this project to the late Ebenezer Kwesi Brace (a.k.a Ceebra)

Contents

Chapter 1

Introduction

Human Activity Recognition (HAR) is an active field of research in machine learning. HAR can be defined as "the ability to interpret the human body gesture or motion through sensors and determine the activity" [6]. The recognition of such activities can be used to solve a lot of problems in various fields of organizations such as banks, airports, hospitals etc. [21, 84]. In banks or airports, real time motion systems [10] are used to capture the movements or motion of people. These movements are analyzed all the time in order to help prevent crimes and dangerous activities from happening.

Also, HAR is used extensively in the health sector for treatment and prevention of several diseases [101]. It has been used in monitoring and treatment of chronic diseases in elderly people [20, 9]. It has also been used to encourage physical exercises in rehabilitation centers for children with motor disabilities [42]. In other areas of health, HAR has been used in estimation of energy expenditure to help in the treatment and prevention of obesity [85].

Human body gesture or motion can be predicted using HAR models with sensor data. Sensors such as gyroscope, accelerometer, LIDAR etc. can detect and respond to inputs from the physical environment [18]. These inputs can be like temperature, light, heat and motion. With the advancement in technology, more and more sensors are embedded in devices for collecting data from human activities [97, 61]. This information recorded by sensors, can be used for predicting the kind of activity which was being performed. Figure 1.1 below shows a general process of HAR. These devices can be placed on multiple body locations, such as the wrist, ankle, chest, or thigh for capturing information of body gesture or motion of a person.

Accelerometer data can also be used in the estimation of energy expended from motion of the body, Energy Expenditure (EE) [9, 42, 76]. EE is also used for treatment and prevention of diseases in the health sector [85, 7, 59, 38].

There are some challenges which arise when analyzing sensor information. For example, multiple activities being performed by different people at different pace and in different environments. One way of tackling such an issue is by using machine learning.

Machine learning can be defined as "a set of methods that can automatically detect patterns in data, and then use the uncovered patterns to predict future data" [80]. Machine learning methods such as Hidden Markov [26], Random Forest [14], Linear Regression [66], Deep Neural Networks [53] etc have been used to build models for HAR [46, 25, 47, 16] and prediction of EE

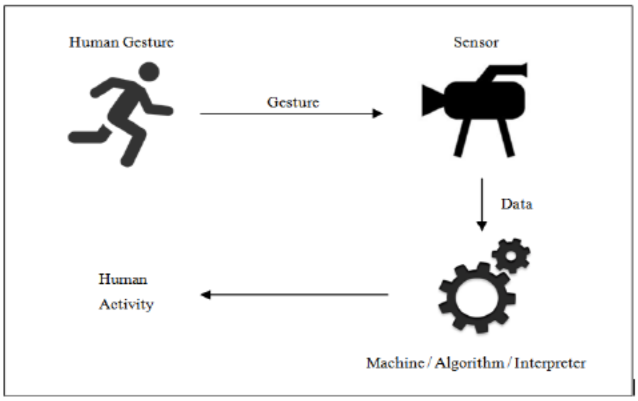


FIGURE 1.1: General process of HAR.

[23, 65, 64]. Mostly because they are able to extract meaningful features and also learn complex relations in large datasets.

Using accelerometer sensor data provided by the Leiden University Medical Center (LUMC), our goal is to develop robust and highly accurate deep neural network models for HAR and prediction of EE. We used Gated Recurrent Units (GRU) and Long Short Term Memory (LSTM) approaches in building these models.

Sometimes there are situations where one or more sensor information is missing [71]. Due to this, our HAR models should be robust and accurate to be able to predict the activity being performed even in the case where some sensor measurements are missing. For that, we use a data augmentation technique based on [86].

We compared our HAR models with previous work using the same datasets. In [71], a feature engineering technique, Accordion [15], was used for constructing features. Besides, in [71] several models were built specifically to detect human activities for one particular sensor or combination of sensor measurements. In our case, we trained a single model for all the cases and without feature construction. Empirical results showed that our method was competitive in detecting the human activities against the several models trained specially for each or combination of sensor measurements.

The rest of the thesis is structured as follows: in Chapter 2, we discuss the related work; in Chapter 3, we discuss the materials and methods used to carry out the research; Chapter 4 we discuss about the experimental setup and results. Finally, in Chapter 5 we present the discussion and conclusion of both HAR and EE research.

Chapter 2

Related Work

2.1 Human Activity Recognition (HAR)

There has been much research in the field of HAR [60, 47, 79, 11, 16]. Due to the recent advancement in sensing technologies, RGB camera-based [100, 93, 83, 68, 81], depth sensor-based [69, 17, 52] and wearable-based sensors [103, 9, 4] are used for collecting sensor information used to train models for HAR.

Several machine learning approaches have been used in HAR [16, 5, 60, 47, 11]. Mannini *et al.* [60] used Hidden Markov Models (HMMs) classifiers to predict human activities from time series accelerometer sensor data. In [47], they compared Hidden Markov Model (HMM) to Conditional Random Field (CRF), Skip Chain Conditional Random Field (SCCRF), and Emerging Patterns for activity recognition from wearable sensors. Also, activity recognition models were develop using Random Forest Classifier from accelerometer sensor data in [16].

In other areas of HAR, Support Vector Machine model was developed for predicting multiclass activity recognition from smart phones [5]. In [11], they used a Kernel Discriminant Analysis for predicting human activity recognition from accelerometer measurements recorded by smart phones. Also, Bayat *et al.* studied the human activity recognition on smart phones using a digital low-pass filter used to isolate the component of gravity acceleration from body acceleration in raw data and evaluated the performance on Support Vector Machines (SVM), Random Forest Classifiers, Multilayer Perceptron, Simple logistic and Logit boost.

2.2 Energy Expenditure (EE)

There has been much research in the field of EE [23, 102, 38, 7, 65, 64]. Sensor information collected from wearable sensor devices are used in various fields such rehabilitation centers, hospitals and residential environments [47, 5, 16, 85].

In rehabilitation centers and residential environments, sensor information collected from wearable devices are used for HAR [9, 42].

In the in hospitals, sensor information collected are used for building machine learning models used for estimation of EE. Several machine learning models have been built for estimation of EE [23, 102, 38, 7, 65, 64].

Dong *et al.* [23] used linear regression and Artificial Neural Networks (ANN) models to compare metabolic energy expenditure estimation from multi-sensor network and single accelerometer.

In [65], they developed an ANN to predict expenditure from wrist sensors, using data collected from 40 adults who performed 14 sedentary, ambulatory, exercise and lifestyle activities.

Also, in [91] they used ANN models to estimate physical activity energy expenditure and identify physical activity type from an accelerometer in.

Rothney *et. al* [82] also used artificial neural network model to estimate the energy expenditure using nonintegrated acceleration signals.

In other areas of EE, Regression models were used in for estimating continuous energy expenditure [99]. Also, Random Forest Classifier was developed in [27] to predict the energy expenditure and type of physical activity from wrist and hip accelerometers.

Chapter 3

Materials and Methods

3.1 Basic concepts of Machine Learning

Machine learning is *"a set of methods that can automatically detect patterns in data, and then use the uncovered patterns to predict future data"* [80]. A machine learning model *"is a model trained to detect patterns in data, and used to make predictions or decisions without being explicitly programmed to perform the task"* [50]. There are 3 types of ML:

- Supervised learning – this is the type of machine learning where we provide the machine learning model with both features and labels to train with; then we use this model to make predictions on unseen data [49]. Example Linear Regression models etc

- Unsupervised learning – this is the 2nd type of machine learning where we provide the machine learning model with only features and the model try to find patterns within such dataset by forming clusters [39]. Example is the KNearest Neigbors (KNN) etc.

- Reinforcement learning – is the 3rd type of machine learning where an object or machine learning model examines its field of environment and gathers information which maximizes the reward or minimizes risk [45].

3.1.1 Regularization

Regularizers are used to prevent overfitting of data during inducting learning phase. Overfitting occurs when a machine learning model learns too much detail in the training data, leading to poor generalization of the model [8, 1]. There are several methods used in preventing overfitting. In neural networks, dropout [89] (See Section 3.8) and early stopping can be used to regularize the models. Early stopping is to stop training the model before the validation loss starts to increase (or the accuracy starts to drop) as shown in Figure 3.1. Also, L2 and L1 [67] are used to prevent overfitting in most machine learning models.

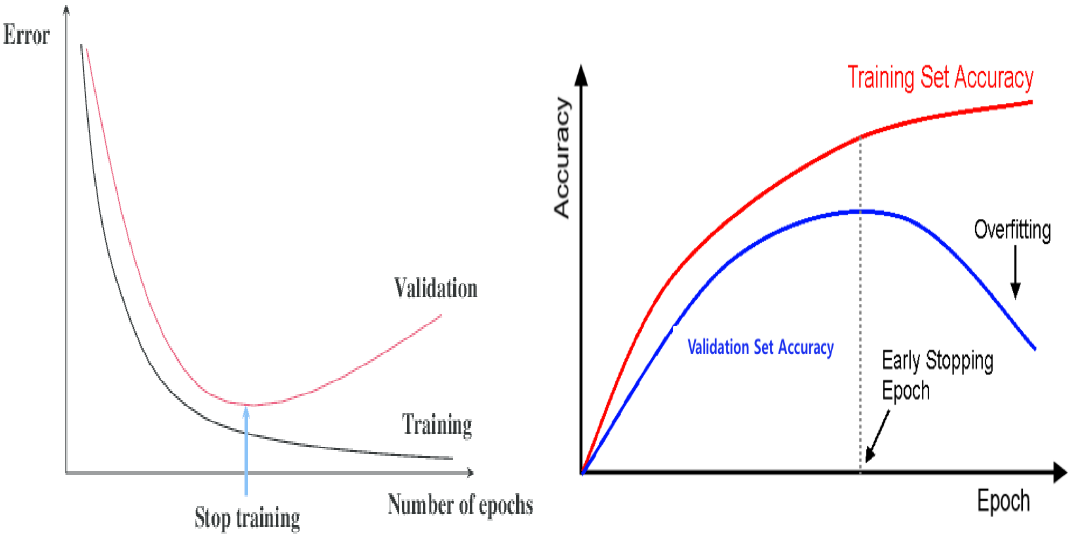


FIGURE 3.1: Example of overfitting during training of machine learning models: (source:(http://fouryears.eu).

3.1.2 Evaluation

In order to evaluate machine learning models, the datasets are typically split into 3 parts, namely, training, testing and validation sets. The training set is used for training the machine learning model. The validation is used to monitor the performance of the machine learning model during training. This is to check if the machine learning model is able to generalize well from the patterns in the training set [88]. Finally, the test set is used to test the performance of the machine learning model when the whole training process is completed.

Other approaches to assess the performance of the models are cross-validation [74], leave-one-out [28] or LOSO is the process where we drop one subject during inductive phase. Then when training is completed, that particular subject is used to test the performance of the model. This process is then repeated again for all the different patients.

Learning curves

Learning curves are used to evaluate the performance of models [94]. In Figure 3.2 we can see an example of learning curves used to evaluate the performance of a model. Learning curves are obtained by computing the error for both training and validation data. There were 2 phenomena, which can be observed with learning curves, that we should try to avoid.

i High Bias – this occurs when both training and validation errors converge at a high point; which results in the model not being able to learn any new information or generalize well on the validation data.

ii High Variance – this occurs when there is a large gap between the training and validation errors. This is due to the model overfitting on the

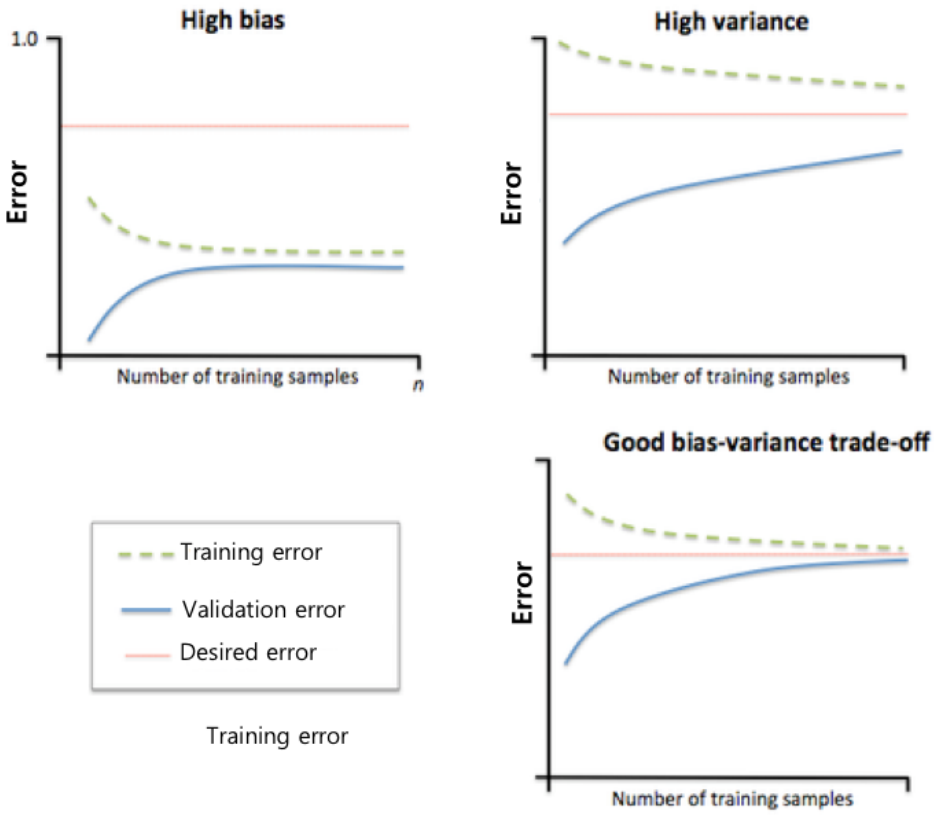


FIGURE 3.2: Learning curves used to evaluate the performance of a model.

training data. One way of rectifying this problem is by training with more data or simplifying the model with less complex features.

Confusion matrix

Confusion matrices are used to illustrate the accuracy of classification models [73]. They are typically made up of 2 dimensions consisting of the *Actual* and *Predicted* classes. The *Actual* classes are usually described in the column headers while the *Predicted* classes are described in the row headers. Example of a confusion matrix is shown in Table 3.1. The following terms are associated with confusion matrices.

1. True positives (TP) – denotes the number of classes which are *true* and the model classifies as *true*.

2. True negatives (TN) – represents the number of classes which are *false* and the model classifies as *false*.

3. False positives (FT) – denotes the number of classes are *false* but the model classifies as being *true*.

4. False negatives (FN) – denotes the number of classes which are *true* but the model classifies as being *false*.

		Actual value		
		p	**n**	**total**
Predicted outcome	p'	True Positive	False Negative	P'
	n'	False Positive	True Negative	N'
	total	P	N	

TABLE 3.1: Example of a confusion matrix.

Evaluation Metrics

Root Mean Squared Error (RMSE) measures the difference between values predicted by a model and values observed [43]. It describes how the model disagrees with the actual data. Lower RSME values show better measure of the model when used in estimating the actual values [43].

R-squared R^2 is the percentage of variation explained by the relationship between independent variables used in prediction of the dependent variable [32].

Accuracy is the total number of correct predictions divided by the total number of predictions:

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

3.1.3 Gradient descent

Gradient descent is an iterative optimization algorithm used in machine learning to find the best local or global minimum of a function [33]. The global minimum is the steepest point of a slope [104]. This is the closest the gradient descent can get in minimizing the cost function [104]. While searching for this global minimum, the algorithm can fall into valleys called local minimum as shown in Figure 3.3. Therefore, to avoid this we train our machine learning models with mini-batch sizes. This computes the gradient on each batch which helps the model to escape such local minimum [33]. Training with mini-batch sizes helps in early convergence of the model [70].

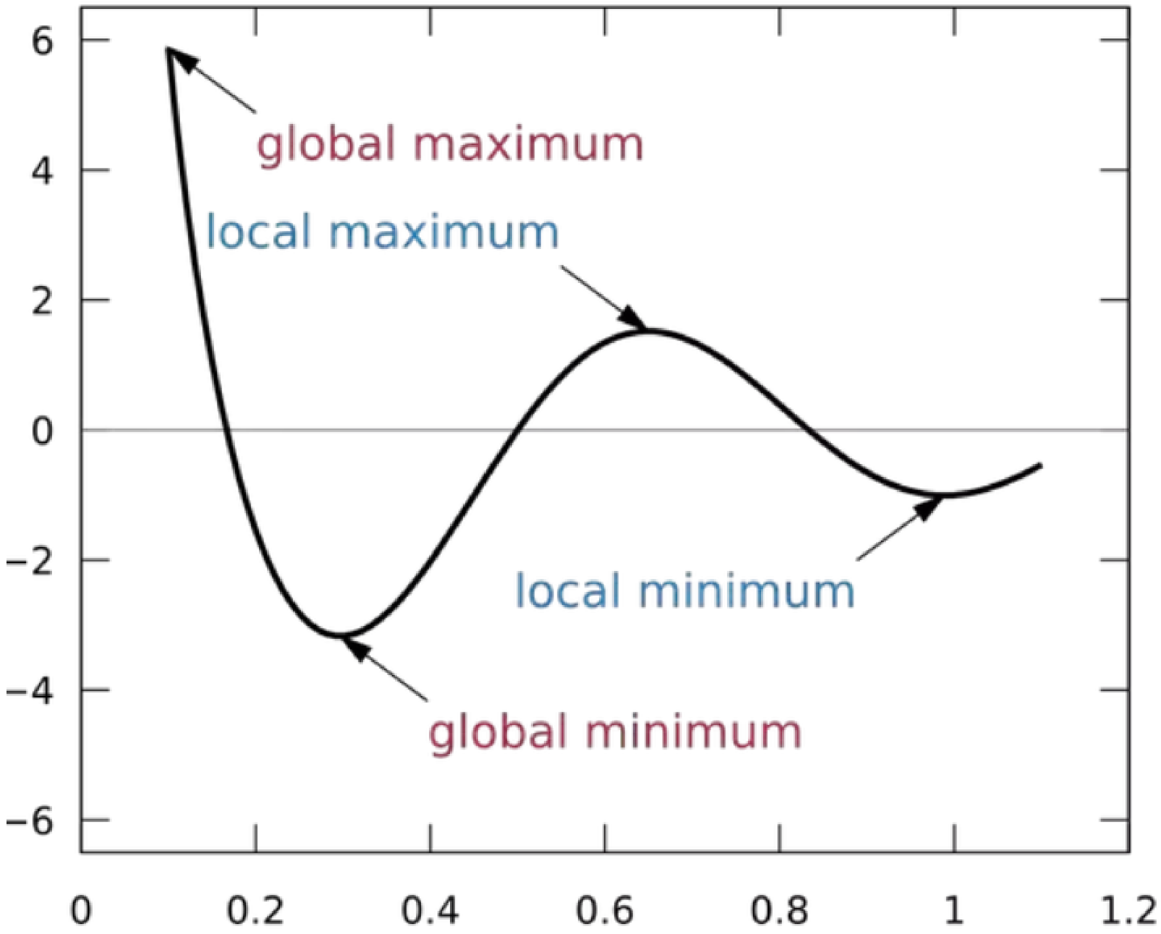


FIGURE 3.3: Diagram showing the global and local minimum of a curve: (source:https://en.wikipedia.org).

3.1.4 Data processing

Resampling

One of the main problems encountered when training a machine learning model for classification is training with an imbalanced number of classes [29, 55]. Imbalance of classes occurs when we have an unequal class distribution, that is, at least one or more of the classes constituting to a small minority. This can lead to the situation where the machine learning models become biased towards the normal classes (majority) when making predictions on unseen data [19].

There has been much literature describing ways to tackle the problem of class imbalance in machine learning [58, 72]. The two commonly used techniques are *oversampling* and *undersampling*.

Undersampling is the process of downsizing the majority class, or classes, to the same number of instances of the minority class [3]. Therefore, using this technique can result in less training data for the model.

Oversampling is the process increasing the number of instances of the minority classes, so they can have more or even the same number of instances as the majority class [44]. One problem associated with this process is that, it might cause the machine learning model to overfit on the training data [44]. One popular oversampling technique, Synthetic Minority Oversampling Technique (SMOTE), was proposed by, Sáez *et al.* [92]. SMOTE uses an iterative ensemble-based noise filter called Iterative-Partitioning Filter (IPF), for oversampling the minority classes to the same size of the majority class.

Feature extraction

Feature extraction is the process of creating new features from initial features. [35]. Some of the reasons why feature extraction is performed is to extract meaningful information from initial features which helps in both the learning and generalization steps of a machine learning model, and sometimes can also be interpreted by humans. Another scenario where feature extraction is performed is the case where the input features to a machine learning model is too large and redundant, hence, smaller subsets of the initial features are constructed and a group of these subsets is selected by feature selection technique. Abdi *et al.* [2] invented a feature extraction algorithm called the Principal Component Analysis (PCA) which is used to generate important features from initial ones by determining the variance and covariance within such features and ranking based on how well they predict on the target variable.

Signal Vector Magnitude (SVM) is a feature extraction technique used for HAR [13]. In order not to confuse it with Support Vector Machines (SVM), we will change the abbreviation to (SMV). SMV "*indicates the degree of movement intensity and is an essential metric in fall detection*" [13]. It is calculated using the equation:

$$SMV = \frac{\sum_{i=1}^{n} \sqrt{x^2 + y^2 + z^2}}{n}$$

Feature selection

Feature selection is the method of selecting relevant subset of features that help to improve the generalization of a machine learning model during inductive learning phase (training of machine learning model) after feature extraction [36]. Selecting which features to use in the inductive learning in order to improve the machine learning model is very challenging, especially in the case of time series sensor data. Even though performing feature extraction helps to create an important subset of features to be used as input data to the machine learning model, it does not help in selecting which features are responsible for better generalization or classification of the machine learning model. Many feature selection techniques have been implemented in this field to help solve such problems. Minola *et al.* [62] used a scoring mechanism to rank algorithm on sample data on the target and the features which are useful for predicting accurately on the target are chosen. Another feature selection algorithm was proposed in [15], named Accordion, that helps to select important features for both classification and regression problems by selecting aggregate features iteratively, in a memory-conscious fashion. Even though these techniques help to select relevant features for inductive learning, they are all based on heuristics.

Downsampling

Downsampling is the method of reducing the sampling rate by an integer factor [78]. Sampling rate is the average number of samples obtained per second. One of the advantages of performing downsampling is to reduce the size of data. Downsampling is used in signal processing [78] etc.

Standardization

Standardization is the process of rescaling the input features to have zero mean and standard deviation of 1 [30]. Standardization can help to speed up the learning process and convergence of the models [30]. Target variables were transformed with *one hot encoding* which converts nominal variables to numerical values.

3.2 Artificial Neural Networks

An Artificial Neural Networks (ANN) [57] are non-linear processing networks that are capable of learning complex relationships within data efficiently. They contain non-linear hidden layers which are assigned weights and biases which learns complex relationships from its inputs and produce an output when an activation function is applied to it.

ANN have been used to solve wide variety of tasks, including human activity recognition [6], speech recognition [77], energy expenditure prediction [7] and image classification [51]. One of the two most commonly type of ANN are *Feedforward Neural Networks (FNN)* (multilayer perceptron [31, 48]) and *Recurrent Neural Networks (RNN)* (Long short-term memory (LSTM)[40], Gated recurrent units (GRU) [22]). FNN [63] are directed graphs with no feedback loop within their hidden layers. The process information straightforward from their input layers to the output layers. RNN [22] contain feedback loops within their hidden layers whose activation at each time depends on that of the previous layer. Figure 3.4 shows an example of a feedfoward neural network.

ANN are trained by iteratively providing the network in batches of data during a certain number of epochs. An epoch is one forward and backward pass of the entire training data through the machine learning model [70]. After every batch, the network computes the generalization error of the model, then adjust its weights to reduce the errors through backpropagation [37].

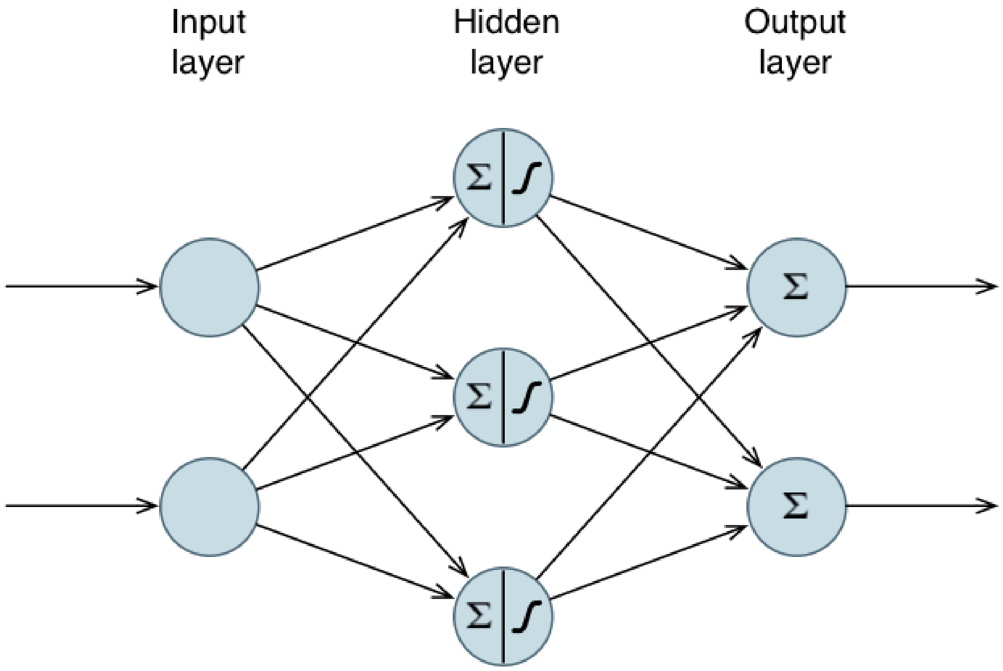


FIGURE 3.4: Feedforward neural network: (source:(https://cs.standford.edu).

3.2.1 Recurrent Neural Networks

Recurrent neural networks (RNN) [22] contain feedback loops within their hidden layers whose activation at each time depends on that of the previous layer. Because of this, they are good architectures to use when dealing with sequential data. RNN have been used to solve a variety of problems, such as language modeling [56], speech recognition [54] etc. Figure 3.5 shows an example of an recurrent neural network.

Even though RNN are able to keep information from sequences, they are not able to do this efficiently when the gap between such sequences becomes too big (due to the vanishing gradient problem [41]). To avoid that, Long short-term memory (LSTMs) and Gated recurrent units (GRUs) neural networks can be used.

Long short-term memory (LSTM)

Long short-term memory (LSTM), is a type RNN that contain memory cells [40, 24]. This allows them to store long range contextual information from sequences [24]. Gates control which information goes through the LSTM model. For that, they use a sigmoid neural network layer and a pointwise multiplication operation. The sigmoid function can transform any values into the interval 0 to 1. When 0, the model forgets the information, when it is 1 it keeps it; they contain three gates, namely, *forget*, *input* and *output* [40].

The forget gate controls which information from the memory cell needs to be forgotten. It contains *sigmoid* function which activates to either 0 or 1.

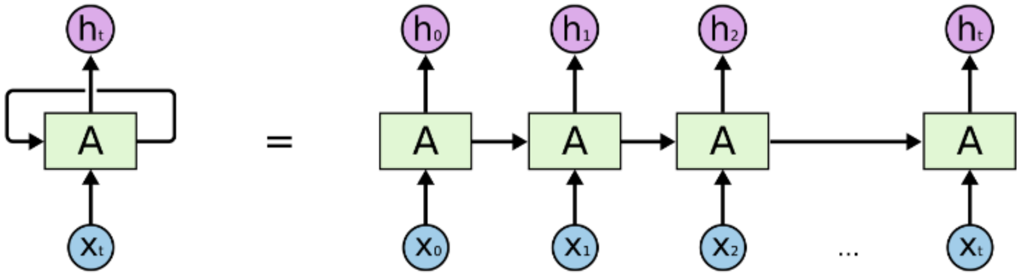


FIGURE 3.5: Representation of a Recurrent Neural Network: (source:http://colah.github.io).

When 0, the model forgets the information, on the other hand, when it is 1 it keeps. The input gate controls which information needs to be updated. Finally, the output gate is used to produce the final outcome. LSTM take into account the information from the previous hidden layer in order to predict the current output. Figure 3.6 shows an example of LSTM network.

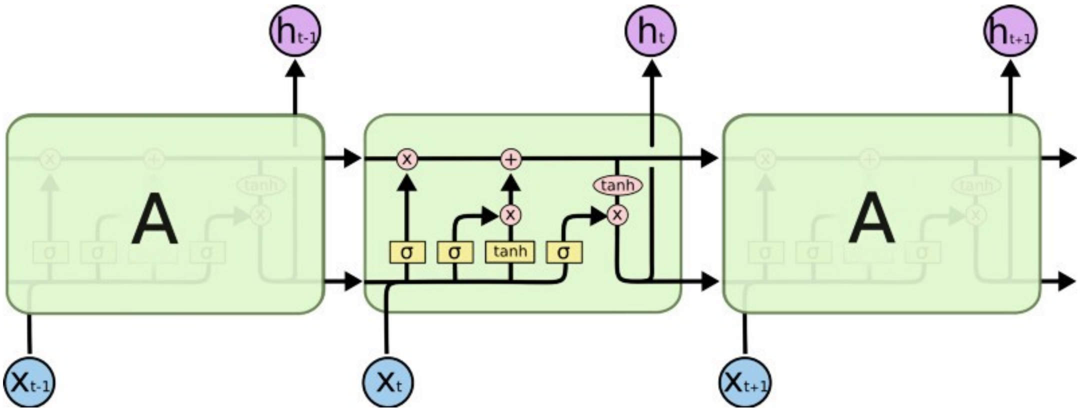


FIGURE 3.6: Long Short Term Memory: (source:http://towardsdatascience.com).

Gated recurrent units (GRUs)

Another type of RNN used for remembering long sequences of information without the vanishing gradient [41, 34] problem are the Gated Recurrent Units (GRU) [22]. Both GRU and LSTM contain memory cells that are used to store information. Unlike LSTMs, GRU have only 2 gates, the *reset* gate and the *update* gate. The reset gates controls which of memory cell information that needs to be forgotten. The update gates controls which amount of information needs to be updated [22, 12]. Figure 3.7 shows an example of GRU network.

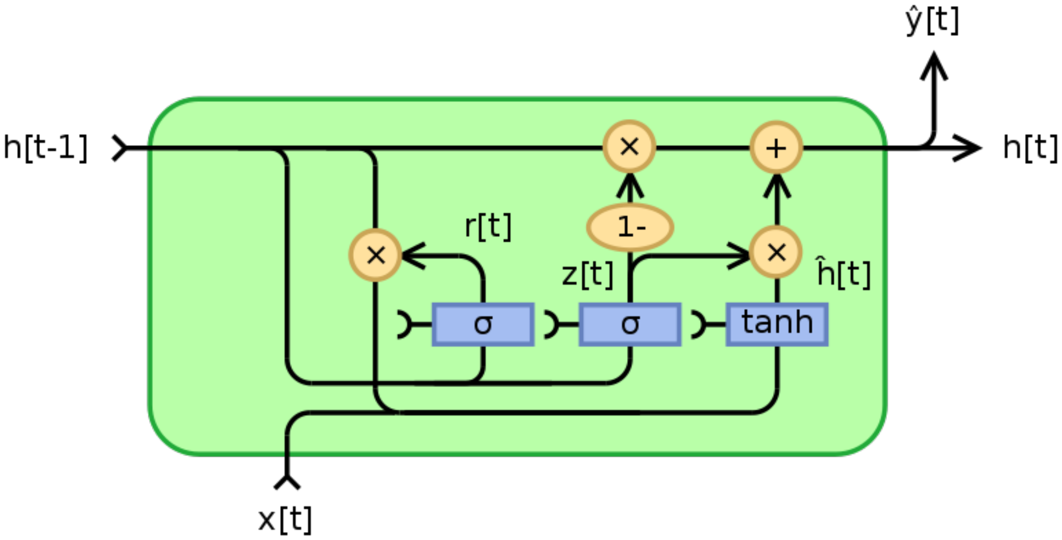


FIGURE 3.7: Gated Recurrent Unit: (source:https://en.wikipedia.org).

3.2.2 Dropout

Dropout is a regularization technique used to prevent overfitting by randomly dropping out neurons (both hidden and visible) in a network during training [89]. The neurons are dropped with a user-defined probability. Training the network with dropout is equivalent as training with different network architectures through extensive weight sharing which are approximately combined together during testing. Networks trained with dropout, helps reduce the generalization error of the networks when used in classification and regression problems [89]. In [89], the authors suggest that 20 and 50 percent of dropouts are often found to be optimal. Figure 3.8 depicts an example of a fully connected network and another network where some of the neurons were dropped.

3.3 Data Augmentation

Data augmentation refers to "*methods for constructing iterative optimization or sampling algorithms via the introduction of unobserved data or latent variables*" [95]. Data augmentation has been used in image classification [51], document analysis [87] etc. It enhances the overall performance of a machine learning model by preventing the model from learning irrelevant patterns during inductive learning phase [95].

The two commonly used data augmentation are, offline data and online data (or on-the-fly) [75]. Offline data augmentation is the process where the data transformations are done before training model. Online data augmentation is when the transformation of data is performed while the model is being trained.

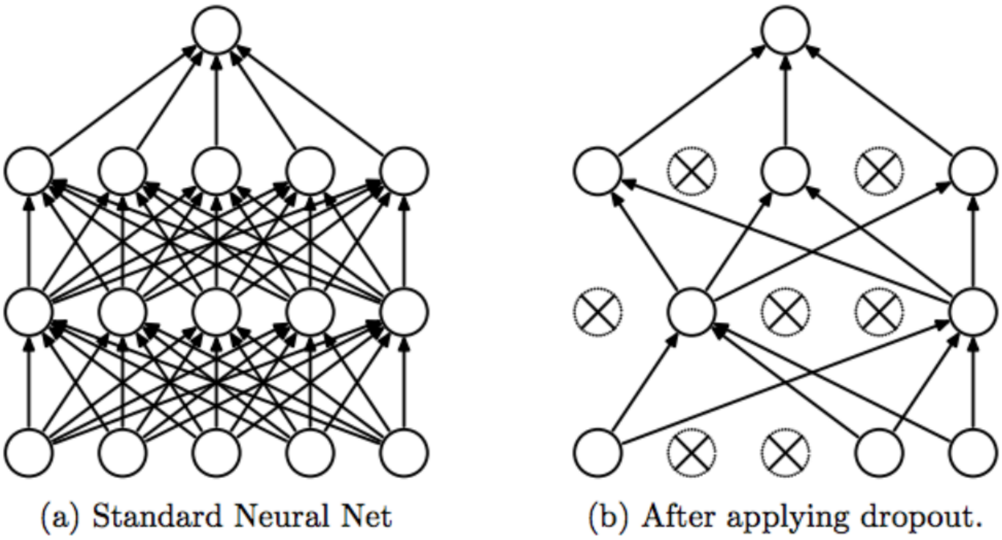


FIGURE 3.8: Dropout technique: (source:https://medium.com).

3.4 Data

The dataset used for this experiment was provided by the Leiden University Medical Center (LUMC) and is referred as GOTOv (Growing Old Together) [71]. The study involved 35 patients (14 females and 21 males) with an average age of 65. Table 3.2 shows the demographic information for all the patients. Each patient performed 16 everyday activities while wearing 6 sensor devices on 6 different body locations. The devices, which are illustrated in Figure 3.9, were:

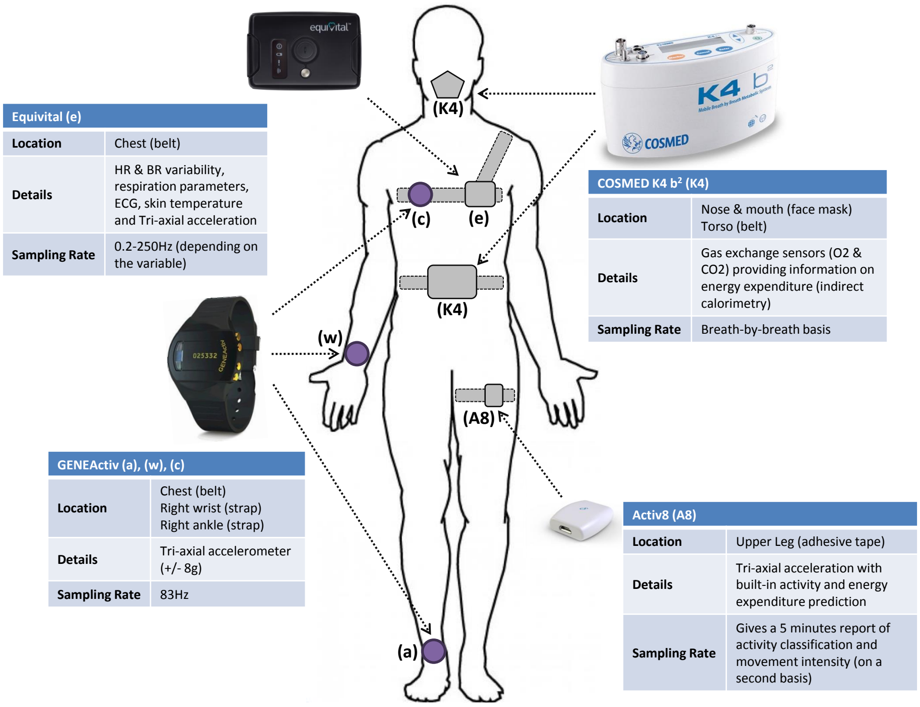
- GeneActiv: placed on the wrist, chest and ankle. Each GeneActiv sensor records triaxial acceleration (+/- 8g) with a high sampling rate of 83Hz.
- COSMED K642: attached to the face through a facial mask and a sensor unit which was used to record the energy expenditure.
- Equivital: was attached to the chest (with a belt) and was used to measure heart rate and heart rate variability, respiration parameters (vivo measurments) and acceleration (tri-axial).
- Polar Electro: It was attached to the chest with a belt which was used to measure heart rate information.
- Philips DirectLife activity monitor: It was placed on the hip and chest with a belt and provided triaxial measurements with a sampling rate of 20Hz.
- Activ8: activity monitor measured acceleration (triaxial), with built-in activity classification. Was placed on the upper leg with a surgical tape.

Figure 3.10 shows the sensor devices used in the GOTOv experiment.

TABLE 3.2: Demographic information for all patients.

Age	Gender	Weight (kg)	Height (cm)
62	female	83	83
66	male	74	177
62	female	75	163
61	female	75	163
59	male	84	177
68	male	91	180
65	male	95	172
64	male	80	172
66	male	84	180
65	male	96	187
60	male	99	190
64	female	66	161
63	male	117	182
69	male	82	182
72	female	74	168
62	female	64	163
59	male	77	180
68	female	70	172
62	male	93	178
62	male	90	182
60	male	83	184
66	female	78	170
70	female	69	160
69	male	85	168
70	female	81	161
64	male	98	179
61	female	82	178
74	male	93	178
67	male	88	174
60	female	70	170
68	female	74	175
68	male	76	175
62	male	77	176
60	male	81	178
81	female	72	167

Because of privacy reasons, we do not present the personal codes of the patients.



Equivital (e)	
Location	Chest (belt)
Details	HR & BR variability, respiration parameters, ECG, skin temperature and Tri-axial acceleration
Sampling Rate	0.2-250Hz (depending on the variable)

COSMED K4 b² (K4)	
Location	Nose & mouth (face mask) Torso (belt)
Details	Gas exchange sensors (O2 & CO2) providing information on energy expenditure (indirect calorimetry)
Sampling Rate	Breath-by-breath basis

GENEActiv (a), (w), (c)	
Location	Chest (belt) Right wrist (strap) Right ankle (strap)
Details	Tri-axial accelerometer (+/- 8g)
Sampling Rate	83Hz

Activ8 (A8)	
Location	Upper Leg (adhesive tape)
Details	Tri-axial acceleration with built-in activity and energy expenditure prediction
Sampling Rate	Gives a 5 minutes report of activity classification and movement intensity (on a second basis)

FIGURE 3.9: Accelerometry sensors placed on a patient.

Every patient involved in the study had to follow a specific protocol (Table 3.3) for the activities performed. Before starting the activities, the COSMED K4b2 sensor was calibrated, which took approximately between 10-15 minutes. The activities were divided into 2 parts, *indoor* and *outdoor*. The indoor activities consisted of *lying down*, *sitting*, *standing* and several household chores, such as *dishwashing*, *stakingShelves* and *vacuumCleaning*. The outdoor activities included the different types of walking *walkingSlow*, *walkingNormal*, *walkingFast* and *cycling*. All patient were allowed a resting period, maximum of 1 minute between the different activities performed, by standing.

Even though each patients was supposed to complete a total of 16 activities, this was not always possible. This was because 7 of the patients could not continue the experiment after completing a certain number of activities. Also, due to weather conditions, most of the outdoor activities were cancelled.

TABLE 3.3: Protocol of the order in which activities are to be performed.

Activity	Duration	Notes
Sensor synchronization	3 minutes	Participant will lightly jump up and down for 20 seconds to synchronize sensor signals
Standing	20 seconds	
Step test	3 minutes	Participant will step up and down a step 20 times at a pace selected by the participant.
Lying down - left	3 minutes	Participant is to turn 90 degrees to the left and remain motionless.
Lying down - right	3 minutes	Participant is to turn 180 degrees to the right and remain motionless.
Sitting sofa	3 minutes	Participant is to be seated and watch TV, browsing channels occasionally.
Sitting couch	3 minutes	Participant is to get seated and read a newspaper.
Sitting desk	3 minutes	Participant is to get seated in the office chair and perform some word processing/browsing.
Ascending stairs	1 minute	Participant is to ascend a single flight of stairs.
Housework dishes	3 minutes	Participant is to wash dishes.
Housework stacking shelves	3 minutes	Participant is to stack shelves with books.
Housework vacuum cleaning	3 minutes	Participant is to perform some cleaning with a vacuum cleaner.
Walking slow pace	5 minutes	Participant is to walk at a slow pace.
Walking medium pace	5 minutes	Participant is to walk at a medium pace.
Walking fast	5 minutes	Participant is to walk at a fast pace.
Cycling	15 minutes	Participant is to cycle at a normal pace

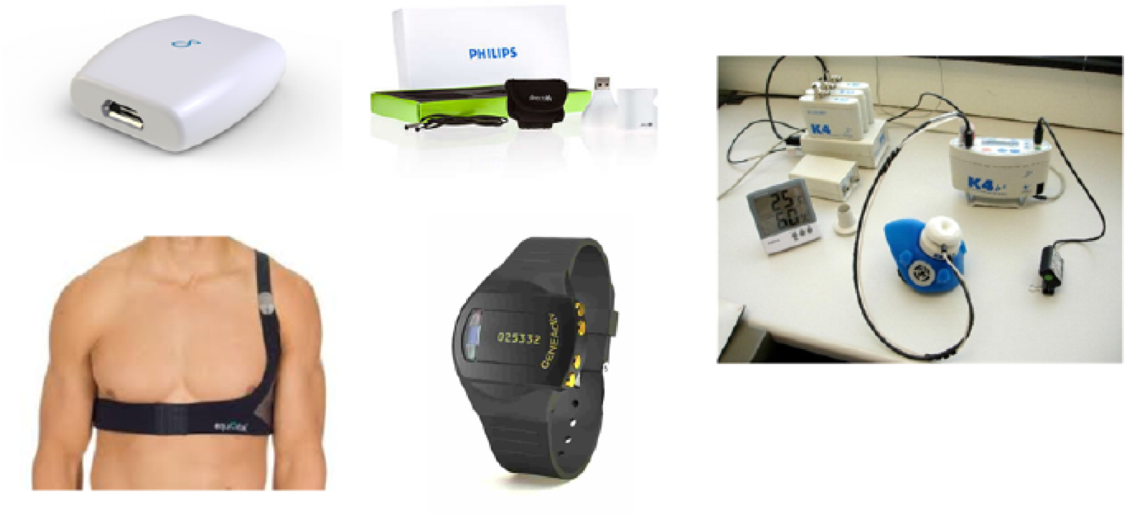


FIGURE 3.10: The devices used in GOTOv study. Active8 and Equivital (left), Philips DirectLife and GENEActive (middle), COSMED K4b2 (right).

Chapter 4

Experimental Setup and Results

In this chapter, we provide information about the experimental procedures we used in conducting our research and is divided in two parts. We discuss about the measures we considered to train models for HAR (Section 4.1) and the prediction of EE (Section 4.2).

4.1 Human Activity Recognition

Six different sensor devices were used in the GOTOv (Growing Old Together) study 3.4. However, for the HAR we only use the sensor measurements recorded by the GeneActiv sensors placed on the wrist, ankle and chest. As mentioned in Section 3.4, each of these sensors records triaxial measurements perpendicularly in the x, y and z axes. Therefore, a total of 9 measurements (or features) are collected per activity (target). In other words, the goal is to predict the correct activity (or target) from the 9 accelerometer sensor measurements.

Not all sensor measurements from the 35 patients was used for training and testing of the models. This was because 7 of the patients could not continue the experiment after completing a certain number of activities (Section 3.4). For this reason, only the data from the remaining 28 patients is used in this study.

To predict human activities from GeneActiv accelerometer data, we use two RNN approaches and a data augmentation technique based on Shekar *et al.* [86].

4.1.1 Data pre-processing

In order to predict the human activities from the original data, several transformations had to be made in the data. We started by standardizing the measurements to zero mean and a standard deviation of 1. Then, due to the choice of the methods, RNN, we had to build sequences of consecutive measurements associated with each activity and each patient. A sequence *"is a finite/infinite list of terms arranged in a definite order, that is, there is a rule by which each term after the first may be found"* [98].

Every sequence is associated with a specific activity and a specific patient. Because activities were performed sequentially, there were chances of having measurements from two different activities in the same sequence. Therefore if a sequence had only part of the sensor measurements of a certain activity, the whole sequence was dropped. In Algorithm 1, we detail the procedure that was used for transforming data into sequences.

We used a time window of 2.5 seconds based on previous experiments [71]. This resulted in sequences of size 200, because of the frequency at which the sensor measurements were sampled (83Hz).

Besides the previously mentioned transformations, we decided to train our models without performing any feature extraction or feature selection technique. The reason for this was because, one of the advantages of using deep neural networks, as mentioned in Section 3.2.1 is their ability to extract relevant knowledge without feature extraction.

Algorithm 1 Transforming data into sequences.

```
$D$ = Original dataset
$s$ = Sequence size
$P$ = Patients
$Q$ = array of sequences
procedure MAKESEQUENCES($D, s$)
    for each patient $p$ in $P$ do
        $A \leftarrow$ Select all activities of patient $p$ in $D$
        for each activity $j$ in $A$ do
            Select all sensor measurements from $D$ for patient $p$ and activity $j$
            Create sequences of length $s$ and assign the sequences to $Q$
        end for
    end for
    return $Q$                                ▷ return sequences for all patients
end procedure
```

Also, as seen in Figure 4.1, the classes are highly imbalanced. Where *cycling* is the most represented class with 66858 instances and *walkingStairsUp* the minority class with 1286 instances. For this reason, we under-sample the data so that the models are trained with an equal number of instances per class.

4.1.2 Proposed approaches

We designed one RNN architecture to tackle the problem of HAR. This architecture was implemented with two variants, one using LSTM layers and other one with GRU layers. As for selecting the number of hidden layers and neurons, since there is no rule of thumb to decide that [90], several other architectures were also previously tested. This final setting was obtained after a careful testing period of 2 weeks of experimental work. The structure presented in Figure 4.2 refers to the one which obtained the most satisfactory results. It consists of an input layer with 9 neurons, 3 hidden layers with 512 neurons and an output layer. The output layer contains 16 neurons, which

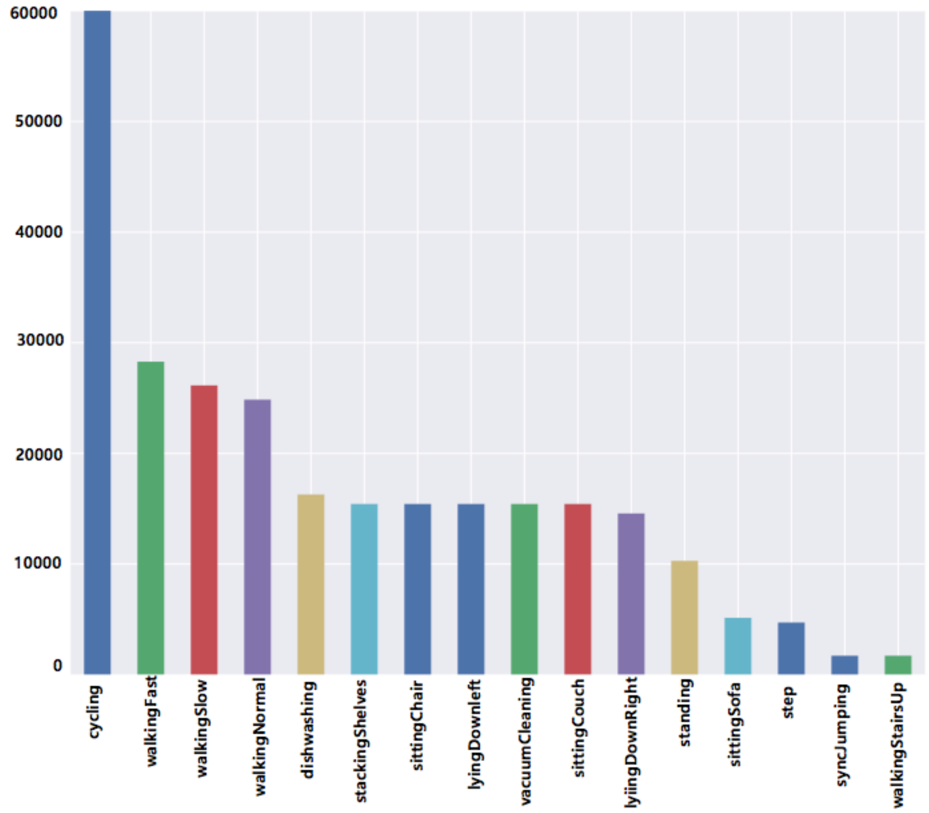


FIGURE 4.1: Distribution of the number of instances per class.

corresponds to the number of targets (Section 3.4). We defined a dropout ratio of 50% between all the layers of the network to prevent overfitting.

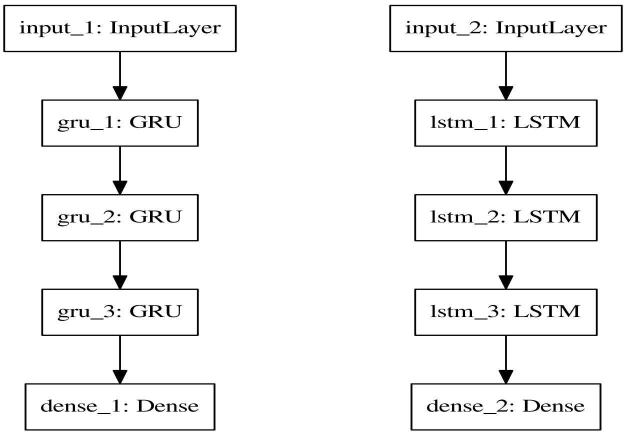


FIGURE 4.2: GRU and LSTM architectures.

Data Generator

The goal of the research was to build one robust and highly accurate model capable of predicting human activities even if some sensor measurements are missing. To achieve that, we used a data *Generator* which included a data augmentation technique [86] to simulate the case where sensor measurements could be missing.

To simulate the case of missing sensor measurements, for every batch, the measurements of a randomly selected sensor, or sensors, had their original values replaced with zeros (Algorithm 2). Once these values are set to zeros, the model is forced to train with the other input features. This helps the model to be more robust to the cases where any sensor is missing.

Algorithm 2 Learning HAR from Missing Data.

```
$q$ = sequence
$K = \{awc, aw, ac, wc, a, w, c\}$
procedure IMPUTESEQUENCES($q, K$)
    $k$ = Randomly select one element from $K$
    if $k == awc$ then
        $T \leftarrow q$
    else if $k == aw$ then
        $T \leftarrow$ set chest values to zeros
    else if $k == ac$ then
        $T \leftarrow$ set wrist values to zeros
    else if $k == wc$ then
        $T \leftarrow$ set ankle values to zeros
    else if $k == a$ then
        $T \leftarrow$ set chest & wrist values to zeros
    else if $k == w$ then
        $T \leftarrow$ set chest & ankle values to zeros
    else if $k == c$ then
        $T \leftarrow$ set ankle & wrist values to zeros
    end if
    return $T$
end procedure
```

Besides the data augmentation, the generator was also used to solve the problem of class imbalance that was present in our training data (See Figure 4.1). For that, on every batch, the number of classes was automatically balanced (Algorithm 3). Moreover, different combinations of sequences were randomly selected at each batch. Thus, making it difficult for the model to memorize the training data.

4.1.3 Evaluation

We tested the performance of the different models using Leave One Subject Out (LOSO) cross-validation as described in Section 3.1.2. All models were trained for 200 epochs with a mini-batch size of 512.

Algorithm 3 Generator

```
$Q$ = List of sequences
$s$ = batch size
procedure TRANSFORMSEQUENCES($Q, s$)
    $T \leftarrow$ empty list
    Undersample training sequences $Q$ to minimum class size
    Create batches of $s$ sequences
    for each sequence $b$ in batch do
        $T \leftarrow$ IMPUTESEQUENCES()
    end for
    return $T$
end procedure
```

To get the overall performance of the proposed approaches, we trained 28 different models using LOSO and then averaged the accuracies for the different models. We trained the models with 25 patients and validate the model on 2 patients. The patients used for the validation were chosen randomly. The data of the remaining patient, left out, was then used to test the model after training was completed.

For training and testing each of the 28 different models we divided our dataset into 3 parts, namely, train, validation and test (as discussed in Section 3.1.2). The training set was used for training the model. The validation set, which included two randomly picked patients, was used to monitor the accuracy and save the best model during training. Finally, the the test set, which included data from only one patient, was used for testing the performance of the model.

We tested how the 28 different models performed on 7 different cases (or scenarios). They were:

awc sensor measurements from all the 3 sensors (*ankle, wrist, chest*).

aw sensor measurements from the ankle and wrist.

ac sensor measurements from the ankle and chest.

wc sensor measurements from the wrist and chest.

a sensor measurements from the ankle.

w sensor measurements from the wrist.

c sensor measurements from the chest.

Also, we computed the t-test score for the 7 different cases. T-test score computes the average between two means and check for significant differences (or importance) between them [96]. We used a package in python called *ttest_ind* to compute such scores.

Finally, we compare the performance of our approaches with previous work Random Forest (RF) where one model was trained for each of the scenarios described above. In other words, the RF models were trained and tested specially for each different combination of sensors: *awc*, *aw*, *ac*, *wc*, *a*, *w*, *c*.

4.1.4 GRU approach

The box and whisker plot in Figure 4.3 gives an overview of the accuracies obtained from the 28 the models trained with the GRU approach in all the 7 different scenarios. In the case of having measurements from all the 3 sensors, *awc*, the models were able to recognize the activities with a minimum and maximum accuracy of 72% and 97% respectively. These results are quite impressive, considering that building models for HAR is a very challenging task, especially in the case where the number of activities are high.

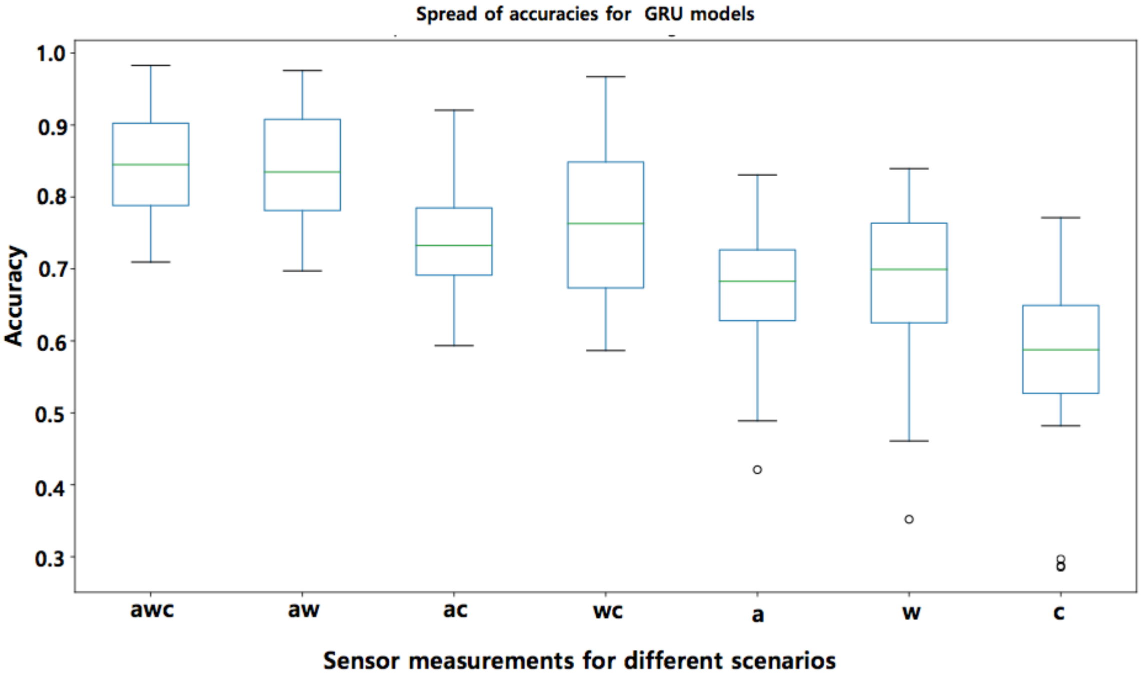


FIGURE 4.3: Spread of accuracies for different scenarios where a sensor could be missing for the GRU models.

In the scenarios where these models only had sensor measurements from two sensors, *aw*, *ac* and *wc*, as expected, the accuracy was affected. In particular for the scenarios *ac* and *wc*, the range of accuracy was 58% – 92% and 57% – 96% respectively. The exception, is the case where the model had no measurements from the chest sensor, *aw*, in which case the accuracy was almost not affected (69% – 98%). The latter can be explained by the fact that the studied activities, have less variation in the chest, and more in the legs and arms.

Finally, when using sensor measurements from only one sensor, *w*, *a* and *c*, the models performed even worst in recognizing the activities. In particular when using only the chest sensor measurements as compared to the case of wrist and ankle the difference was more striking. This can due to little changes in the measurements recorded by the chest sensor as already mentioned above. Therefore, resulting in the models predicting the activities with accuracies of 52% – 76%, as compared to the 46% – 86% and 49% – 80% for the wrist and ankle measurements respectively. Moreover, we would like to highlight that some of these models predicted the activities with very low accuracy (see Figure 4.3) represented by the outliers.

The results presented in Figure 4.4 show that our method is competitive in detecting the activities against the several models of Random Forest (RF) trained specially for each or combination of sensor measurements. The barplot in Figure 4.4 shows the average accuracy of both RF and GRU models obtained in the different scenarios. The overall average accuracy for both GRU and RF models were 74.6% and 74.5% respectively.

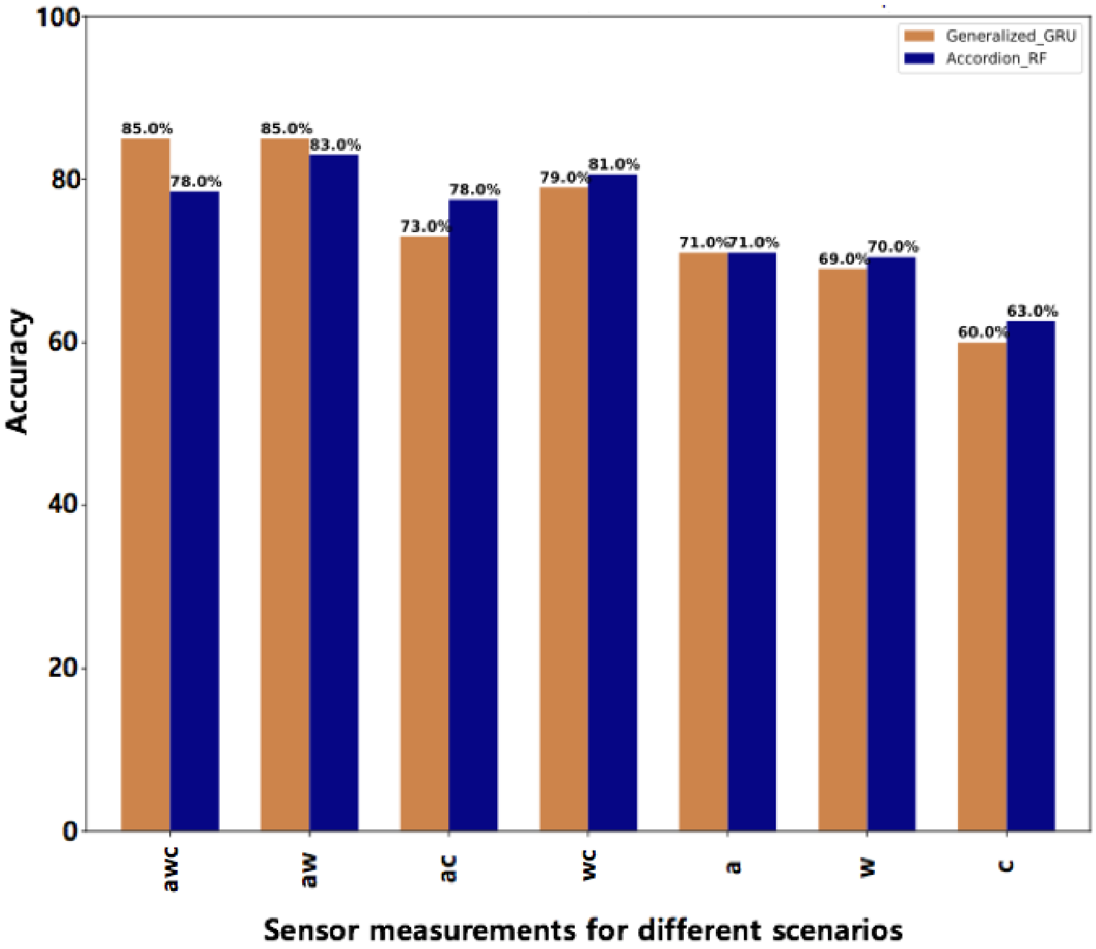


FIGURE 4.4: An average accuracy of the GRU and RF models for the different scenarios. The x-axis represents the different scenarios (*awc, ac, aw, wc, a, c, w*) and the y-axis the average accuracy.

With all measurements from 3 sensors present, *awc*, both GRU and RF models were able to predict the activities with an average accuracies of 85% and 78% respectively. A t-test showed that the GRU models are statistically significant better for the recognition of activities in the scenario where all the sensor measurement are provided.

Having sensor measurements from only two sensors, *aw*, *wc* and *ac*, GRU models were able to predict the activities with average accuracies of 85%, 79% and 73% respectively. On the other hand, models trained with RF approach were able to predict the activities with average accuracies of 83%, 81% and 78% respectively. However, this time there was no significant differences between the GRU and RF models.

From Figure 4.4 in can be observed that both GRU and RF models predicted the activities with similar accuracy for the scenario where we had sensor measurements from only one sensor. The average accuracies were (71%, 69%, 68%) and (71%, 70%, 63%) for the GRU and RF methods respectively. Also, the t-test score for the different scenarios, showed no significance between the two models when used for HAR.

Ankle, wrist and chest confusion matrix

Figure 4.5 shows the confusion matrix of the predictions of GRU models, in the senario of having measurements from all the 3 sensors, *awc*. We could observe that the models were able to detect the various household (*dishwashing, stakingShelves, vacuumCleaning*), sitting (*sittingChair, sittingCouch, sittingSofa*), lying down (*lyingDownLeft, lyingDownRight*) and walking (*walkingSlow, walkingFast, walkingNormal*) activities.

In terms of missclassifcation, it seems to be higher between *dishwashing* and *stakingShelves*; *vacuumCleaning* and *standing*; and the different walking (*walkingSlow, walkingFast, walkingNormal*) activities. In the case of the activities *vacuumCleaning* and *standing*, since the patients while vacuuming might take breaks (by standing for a minute or more). This can explain the high number of *vacuumCleaning* classified as *standing* (Figure 4.5). Therefore, resulting in both activities having similar sensor measurements recordings which leads to the confusion. In terms of *dishwashing* and *stakingShelves* activities, the confusion could result from the pace at which each patient performed the activity.

Also, the misclassification between the different types of walking (*walkingSlow, walkingFast, walkingNormal*) might result of the different walking pace for each individual. A person moving at a fast pace might be a slow pace for another person. This resulted in different variations in the sensor measurements recorded by different patients for the same activities.

Ankle and wrist confusion matrix

In the case of having measurements from the sensors in the ankle and wrist, *aw*, the models were able to predict the activities with similar results as of the case of having measurements from all the 3 sensors, *awc*. As in the case of *awc*, the models were also capable of distinguishing between the various household, walking, lying down and sitting activities as seen Figure 4.6.

However, without the chest sensor measurements, the number of misclassified examples between the different types of walking activities almost doubled if compared to the scenario of having all the sensor measurements

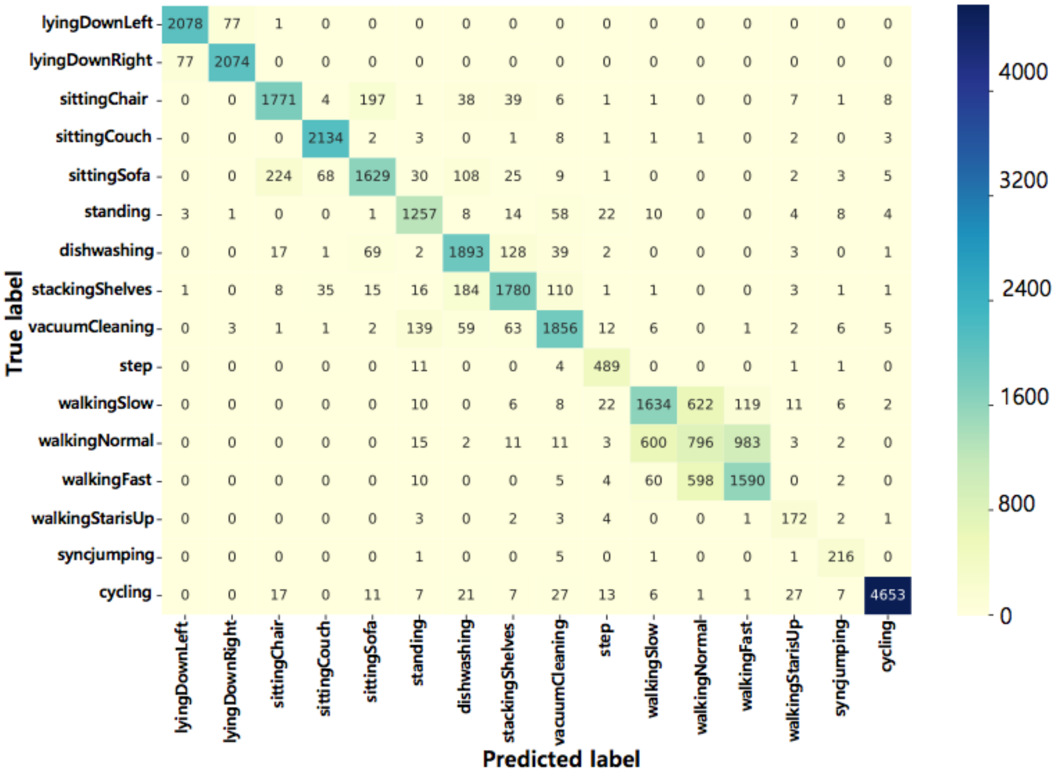


FIGURE 4.5: Confusion matrix of the predictions from all models having ankle, wrist and chest sensor measurements.

from the 3 sensors. Furthermore, there was an increase in misclassification between the different types of lying down activities as shown in Figure 4.6.

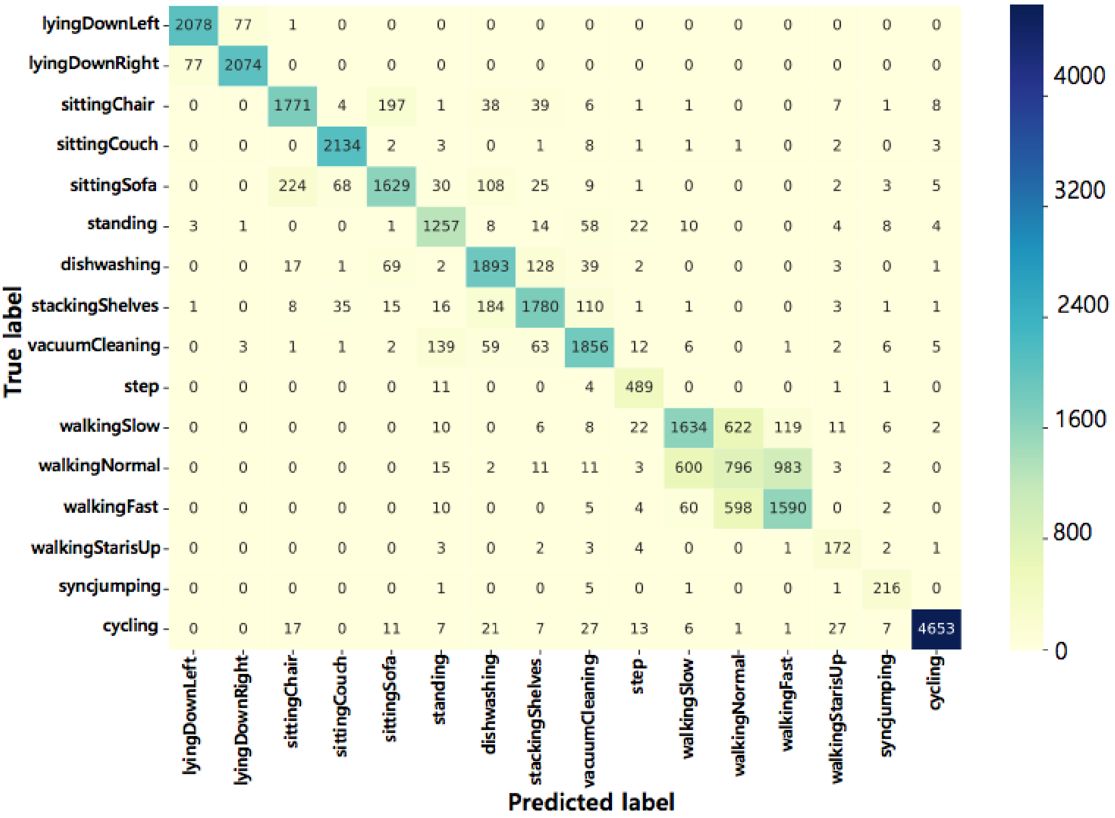
In addition, in the other two cases, where sensor measurements from the ankle and chest *ac* or wrist and chest, *wc*, were used, the models were better in recognizing the different household activities in the latter (see Figure A.1 and Figure A.2) for confusion matrices).

Finally, having only the ankle and chest measurements, *ac*, the models performed bad in detecting the various household activities when compared to the *aw* and *wc* cases. In contrast, they were better at recognizing the different classes of sitting and walking activities. This can be explained by more variation in measurements recorded by the ankle and chest sensors when a patient was performing such activities.

Ankle confusion matrix

Finally, we analyse the predictions of the GRU models obtained from sensor measurements of the ankle only, i.e. scenario *a*. In Figure 4.7 represents the confusion matrix for all 28 patients. It can be observed that, the models were as good as detecting the difference between the lying down and sitting activities.

Even though with measurements from only the sensor on the ankle, the models were still able to distinguish many activities, such as *cycling* and *lyingDown*. The models had a high number of misclassified instances between



	lyingDownLeft	lyingDownRight	sittingChair	sittingCouch	sittingSofa	standing	dishwashing	stackingShelves	vacuumCleaning	step	walkingSlow	walkingNormal	walkingFast	walkingStarisUp	syncjumping	cycling
lyingDownLeft	2078	77	1	0	0	0	0	0	0	0	0	0	0	0	0	0
lyingDownRight	77	2074	0	0	0	0	0	0	0	0	0	0	0	0	0	0
sittingChair	0	0	1771	4	197	1	38	39	6	1	1	0	0	7	1	8
sittingCouch	0	0	0	2134	2	3	0	1	8	1	1	1	0	2	0	3
sittingSofa	0	0	224	68	1629	30	108	25	9	1	0	0	0	2	3	5
standing	3	1	0	0	1	1257	8	14	58	22	10	0	0	4	8	4
dishwashing	0	0	17	1	69	2	1893	128	39	2	0	0	0	3	0	1
stackingShelves	1	0	8	35	15	16	184	1780	110	1	1	0	0	3	1	1
vacuumCleaning	0	3	1	1	2	139	59	63	1856	12	6	0	1	2	6	5
step	0	0	0	0	0	11	0	0	4	489	0	0	0	1	1	0
walkingSlow	0	0	0	0	0	10	0	6	8	22	1634	622	119	11	6	2
walkingNormal	0	0	0	0	0	15	2	11	11	3	600	796	983	3	2	0
walkingFast	0	0	0	0	0	10	0	0	5	4	60	598	1590	0	2	0
walkingStarisUp	0	0	0	0	0	3	0	2	3	4	0	0	1	172	2	1
syncjumping	0	0	0	0	0	1	0	0	5	0	1	0	0	1	216	0
cycling	0	0	17	0	11	7	21	7	27	13	6	1	1	27	7	4653

FIGURE 4.6: Confusion matrix of the predictions from all models having ankle and wrist sensor measurements.

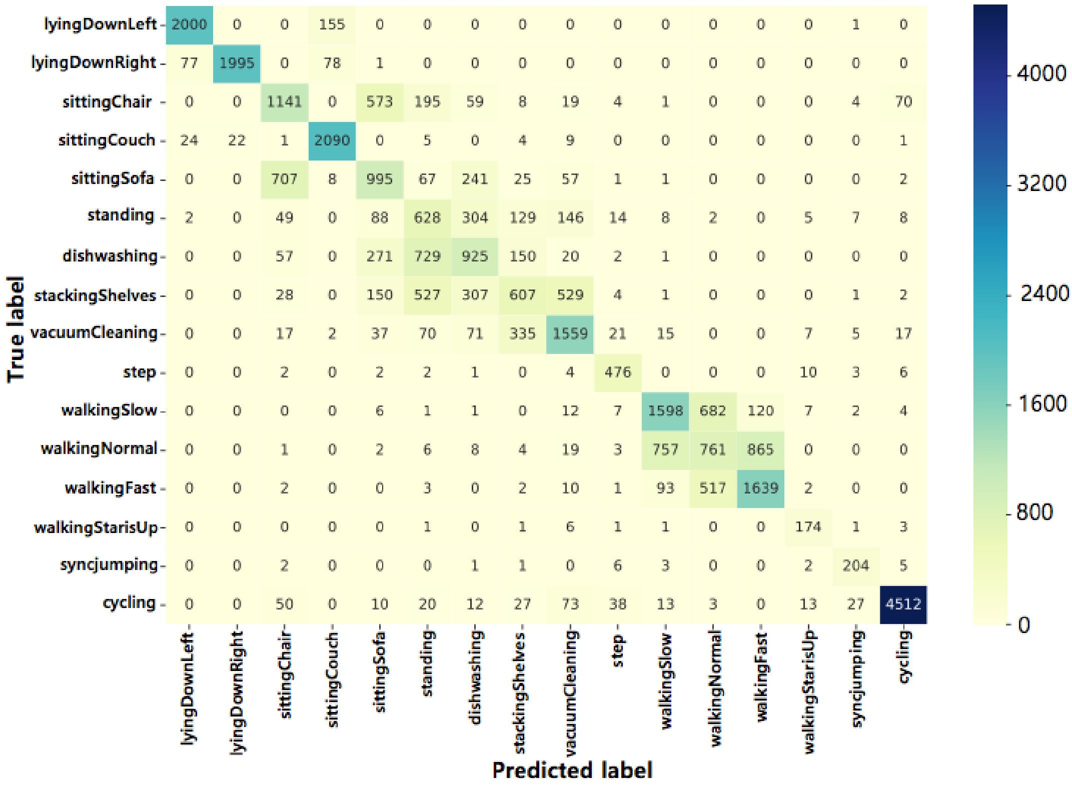
the different household activities. For example, in the case of *dishwashing* and *Standing* activities; the *Standing* activity involves similar use of the legs as compared to *diswashing*, which on the other hand, makes more use of the hands.

In the other two cases, w or c, the models were able to distinguish a bit better for the various household activities (see Figure A.4 for confusion matrices). The misclassification was still quite high for the different sitting and walking activities when compared to the ankle sensor.

Lastly, using the models to predict the human activities in the scenario where we had chest measurements only, c, the results were worst when compared to the case of having measurements from the wrist or ankle. However, the chest sensor was better in recognizing the different lying down activities. This might be due to more movement of the chest when performing such activities. In conclusion, the wrist and ankle sensors were better than the chest sensor when used for HAR in this study.

4.1.5 LSTM approach

The barplot in Figure 4.4 represents the average accuracy of both LSTM and GRU models obtained in the different scenarios. As mentioned before, both networks were trained with the same settings. Besides the better accuracy, during training, we observed that the models trained with GRU layers were



	lyingDownLeft	lyingDownRight	sittingChair	sittingCouch	sittingSofa	standing	dishwashing	stackingShelves	vacuumCleaning	step	walkingSlow	walkingNormal	walkingFast	walkingStarisUp	syncjumping	cycling
lyingDownLeft	2000	0	0	155	0	0	0	0	0	0	0	0	0	0	1	0
lyingDownRight	77	1995	0	78	1	0	0	0	0	0	0	0	0	0	0	0
sittingChair	0	0	1141	0	573	195	59	8	19	4	1	0	0	0	4	70
sittingCouch	24	22	1	2090	0	5	0	4	9	0	0	0	0	0	0	1
sittingSofa	0	0	707	8	995	67	241	25	57	1	1	0	0	0	0	2
standing	2	0	49	0	88	628	304	129	146	14	8	2	0	5	7	8
dishwashing	0	0	57	0	271	729	925	150	20	2	1	0	0	0	0	0
stackingShelves	0	0	28	0	150	527	307	607	529	4	1	0	0	0	1	2
vacuumCleaning	0	0	17	2	37	70	71	335	1559	21	15	0	0	7	5	17
step	0	0	2	0	2	2	1	0	4	476	0	0	0	10	3	6
walkingSlow	0	0	0	0	6	1	1	0	12	7	1598	682	120	7	2	4
walkingNormal	0	0	1	0	2	6	8	4	19	3	757	761	865	0	0	0
walkingFast	0	0	2	0	0	3	0	2	10	1	93	517	1639	2	0	0
walkingStarisUp	0	0	0	0	0	1	0	1	6	1	1	0	0	174	1	3
syncjumping	0	0	2	0	0	0	1	1	0	6	3	0	0	2	204	5
cycling	0	0	50	0	10	20	12	27	73	38	13	3	0	13	27	4512

FIGURE 4.7: Confusion matrix of the predictions from all models having ankle sensor measurements.

faster when compared with models with LSTM layers. It took 22 seconds to complete an epoch using GRU models, while it took 222 seconds in the case of LSTM models. The overall average accuracy for both GRU and LSTM models were 74.6% and 64.9% respectively.

For both models, there can be seen a decline in the accuracy (see Figure 4.4) when sensor measurements are removed. However, the relative decrease in percentage between the LSTM models and the GRU models for the different scenarios (*awc, aw, ac, wc, a, w, c*) was 6%, 7%, 10%, 7%, 11%, 12% and 15% respectively. This seems to indicate that GRU layers deal better with missing data as compared with LSTM layers. A t-test score showed that there was a significant difference between both models (LSTM and GRU) for all the different scenarios. Furthermore, observing the box and whiskeys plot in Figure 4.9, the worst and best performance of the GRU models in HAR outperformed that of the LSTM models.

4.2 Energy Expenditure

In this experimental part we use sensor measurements recorded by the *Cosmed K4b2* and *GeneActiv* sensors placed on the wrist and ankle. Besides that we also used demographic information for training models to predict the energy expenditure (EE). As mentioned in Section 3.4, the *Cosmed K4b2* sensor device is used to record the EE. Therefore, our goal is to predict the *Cosmed K4b2* EE

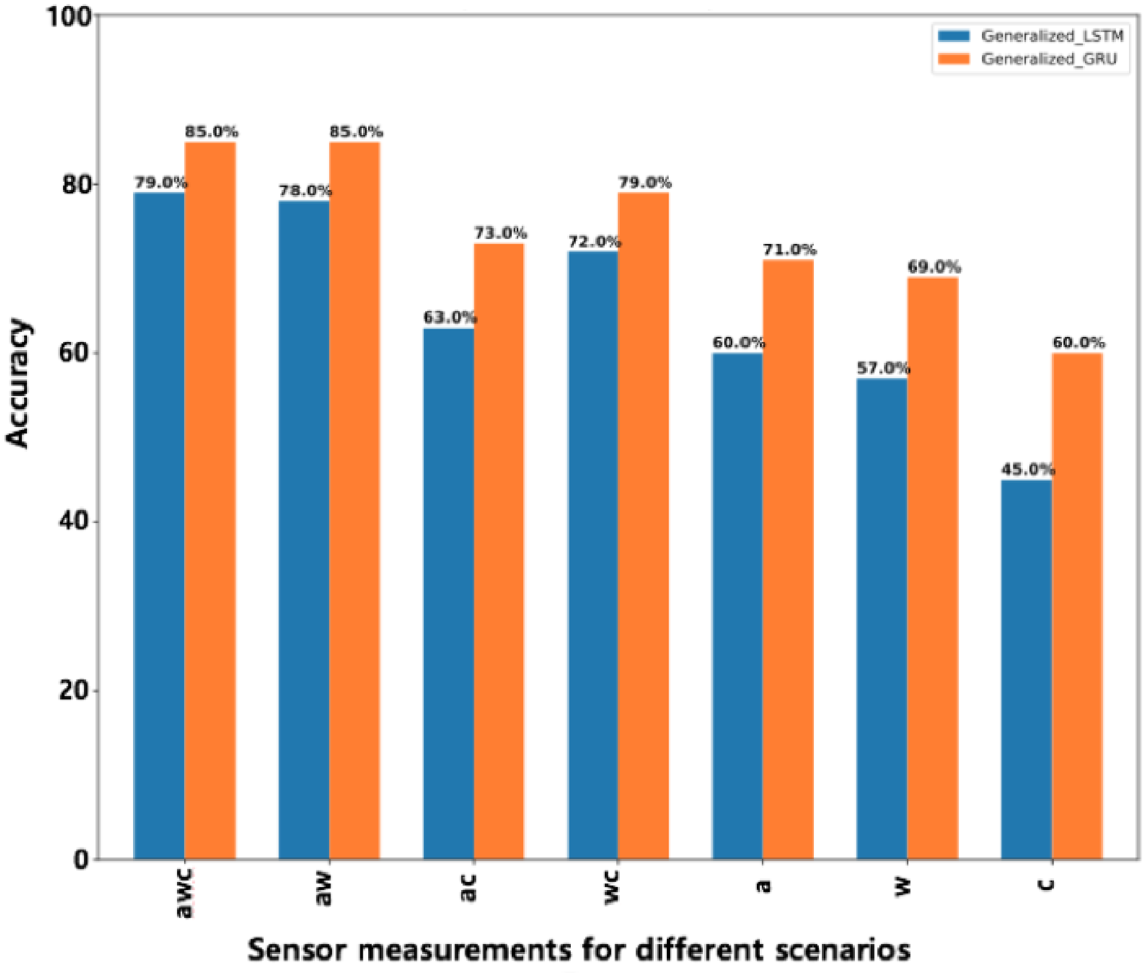


FIGURE 4.8: Barplot showing the average accuracy of the LSTM and GRU models for the different scenarios. The x-axis represents the different scenarios (*awc, ac, aw, wc, a, c, w*) and the y-axis the average accuracy.

from sensor measurements collected from the *GeneActiv* together with demographics information.

4.2.1 Proposed approach

The architecture of the neural network used to predict the EE was made up of one recurent GRU and one feedforward networks, combined into one final feedforward network (Figure 4.10). This was because, the datasets used for training the models consisted of both time series (*sensor measurements*) and static (*demographics*) data. Therefore, we built sequences for the sensor measurements and used them to train the recurrent network with GRU layers. At the same time, we feed the static data to the feedforward network.

Since there is no rule of thumb when it comes to selecting the number of hidden layers and neurons, several other settings were also previously

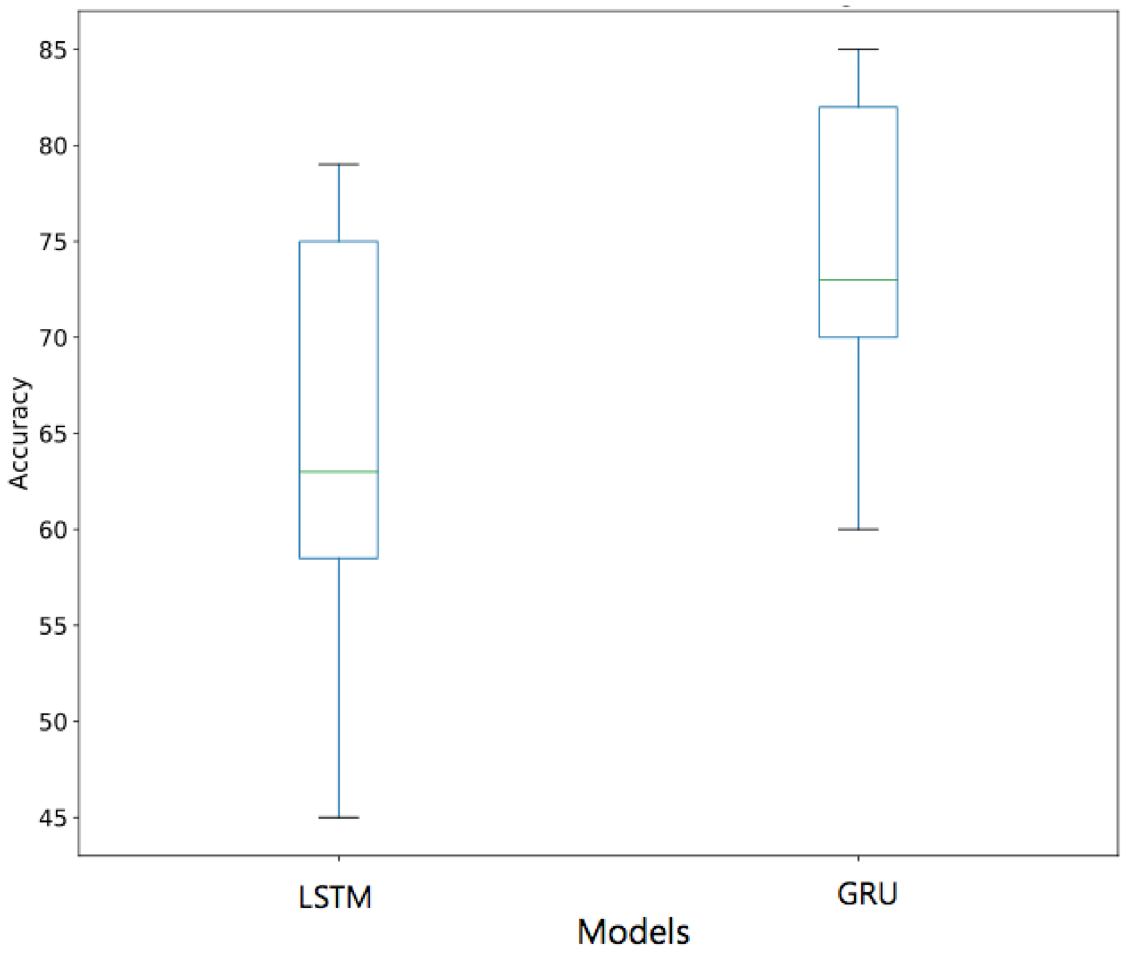


FIGURE 4.9: Box plot showing the overall performance of the GRU and LSTM models.

tested. The presented structure in Figure 4.10 refers to the one which provided the most satisfactory results. The RNN architecture consisted of an input layer, 3 hidden layers and an output layer. The input layer had 9 neurons, while the neurons in the hidden layers were 32, 256, 32, 32 respectively. The feedfoward network had an input layer with 4 neurons and a hidden layer with 32 neurons. The output layers of both networks were concatenated and connected to 2 more hidden layers and an output layer consisting of 32, 16 and 1 neurons respectively. The output layer is made up of only 1 neuron which is used to estimate the EE.

We applied a dropout ratio of 50% to all the layers of the recurrent network and 20% to final hidden layers of the network structure. Figure 4.10 shows the described network structure.

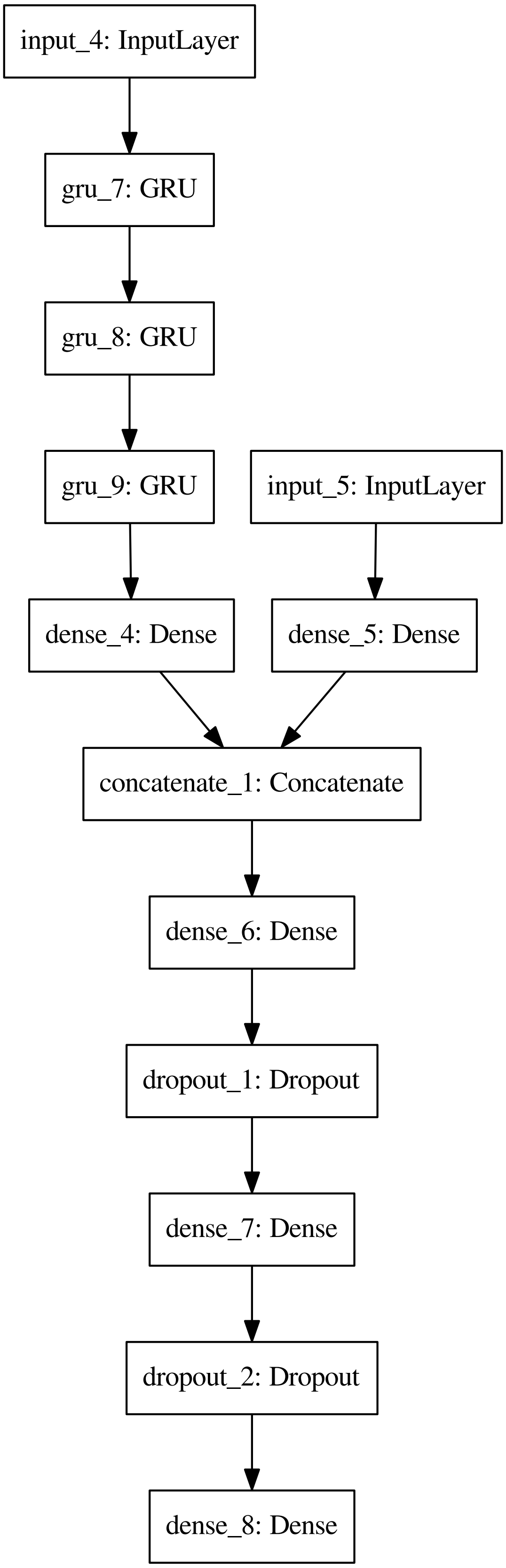


FIGURE 4.10: Neural network architecture for the prediction of Energy Expenditure.

4.2.2 Data pre-processing

Window Size

In order to predict the EE, we had to transform parts of the data into sequences. In this work, a sequence is defined as a 2 dimensional matrix ($\{t, f\}$), where t is the time steps and f is the number of features. At each time step, there are f sensor measurements and each represents one feature.

Uniform sampling rates One problem we encountered when building sequences was in defining a reasonable time window. We experimented with different time windows of 1, 2, 4, 6, 7 and 10 minutes which originated sequences with 4980, 9960, 19920, 34860, 39840 and 49800 time steps respectively. However, when training our models with these sequences, we encountered an insufficient memory problem. This was because, we could not fit all the sequences into memory for training. Therefore, we tested different sequences with time steps from 200-500. Moreover, we downsampled the original data (at 16Hz) to lower sampling rates of 1, 2, 4, 6 and 8 Hz. Finally, we observed that a sequence with 460 time steps (representing a time window of 4 minutes) with a sampling rate of 2 Hz was the most reasonable choice. We transform our data into sequences of dimension (t, f), downsampled to a sampling rate r.

Algorithm 4 Transforming data into sequences.

```
$X$ = *GeneActiv* measurements
$E$ = energy expenditure values
$t$ = number of time steps
$r$ = sampling rate
**procedure** SEQUENCESGENERATOR($X, r, t, E$)
    $w \leftarrow$ empty list
    **for each** e in E **do**
        Create sequence with $t$ time steps from $X$ with sampling rate $r$ where the last value in the sequence represents has the same timestamp as $e$
    **end for**
    **return w**
**end procedure**
```

Non-uniform sampling rates Using the time window of 4 minutes, we performed another experiment where we built sequences using non-uniform sampling rates. We compared whether having more recent data in a sequence, helped improved the estimation of the EE. We built sequences for training such models as follows:

i Take a sequence of 4 minutes of data.

ii Split the sequence into 4 equal segments.

iii Downsample the sensor measurements in each segment using sampling rates 1, 2, 4, and 8.

Algorithm 5 Transforming data into sequences.

```
$X$ = *GeneActiv* measurements
$E$ = energy expenditure values
$t$ = number of time steps
$r$ = sampling rate
procedure SEQUENCESGENERATOR($X, r, t, E$)
    $w \leftarrow$ empty list
    for each e in E do
        Create sequence with $t$ time steps from $X$
        Split measurements in sequence into 4 equal segments
        Downsample the measurements in each segment using sampling rates in $r$
    end for
    return w
end procedure
```

We used sampling rates of 1, 2, 4 and 8 Hz to downsample the measurements in each sequence. We built two different sequences to test our hypothesis. The first sequence w_l, had more recent sensor measurements, while the second sequence w_j, had less recent sensor measurements as shown in Figure 4.11 below. The empirical results showed that having more recent sensor measurements in a sequence helped improved the model in estimation of EE.

We also created features for training our models. We used the *SMV* equation (see Section 3.1.4), to combine the acceleration measurements recorded in 3 triaxial perpendicular axes (*x*,*y*,*z*) into one value.

Hence, we trained 4 different models and compared them. Two of the models were trained with raw sensor measurements while the other two were trained with SMV. We named the 4 different models as follows to enable us easily discuss the results of the models.

1. *flat* - this model was trained with sequences transformed using algorithm 4 above.

2. *SMV_flat* - this model was trained with SMV sensor measurements transformed into sequences using algorithm 4 above.

3. *equalSegments* - this model is trained with sequences transformed using algorithm 5 above.

4. *SMV_equalSegments* - this model is trained with SMV sensor measurements transformed into sequences using algorithm 5 above.

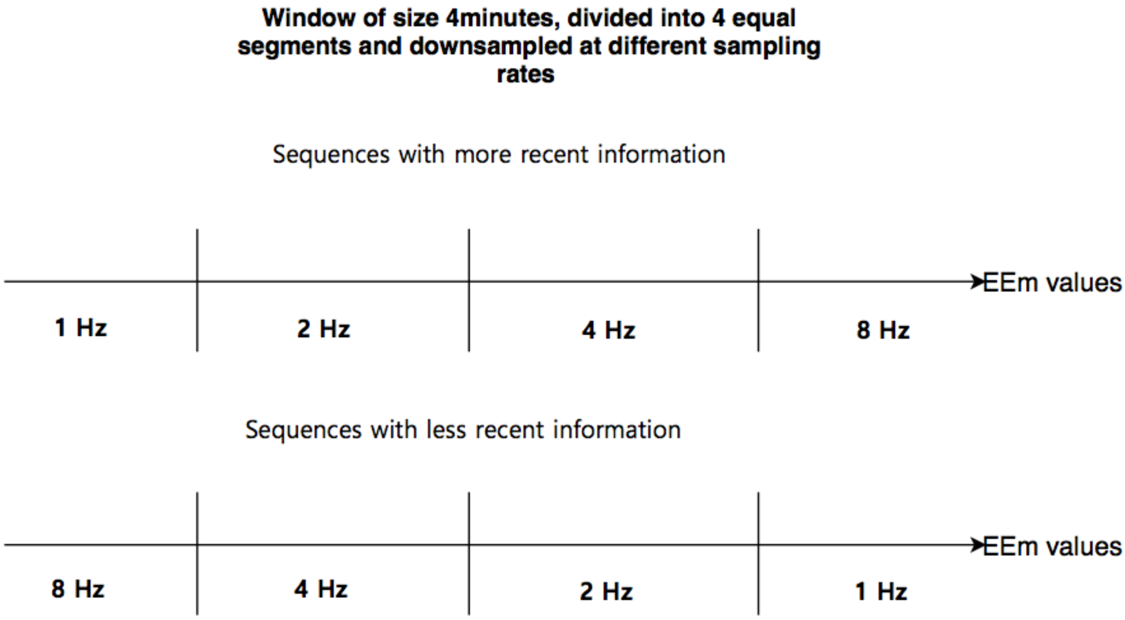

Window of size 4minutes, divided into 4 equal segments and downsampled at different sampling rates
Sequences with more recent information
1 Hz
2 Hz
4 Hz
8 Hz
EEm values
Sequences with less recent information
8 Hz
4 Hz
2 Hz
1 Hz
EEm values

FIGURE 4.11: Equal segment of data measurements downsampled with different sampling rates.

4.2.3 Evaluation

We trained and tested the performance of the models in the 4 different approaches using Leave One Subject Out (LOSO) cross-validation. All models were trained for 200 epochs with a mini-batch size of 1024.

Also, there were cases where there was no Cosmed EE measurements recorded for the activity which was being performed. Therefore we did not include the sensor information from such patients when training our models.

We trained the models with 23 patients, validated the performance of the models during training on 1 patient, and tested the performance of the final model after training on the last patient which was left out. Hence, to get the overall performance of one model, we trained 25 different models using LOSO mentioned in Section 3.1.2. The validation dataset was randomly chosen for all the LOSO models. We tested the performance of the machine learning models by measuring both the Root Mean Squared Error (RMSE) and R-squared (R^2) values for all the models when training was completed.

4.2.4 EE Models

We tested the performance of the different models used in estimating the EE by recording the average RMSE and R^2 values for all the LOSO trained models.

Table 4.1 shows the average LOSO R^2 and RMSE values for all the different machine learning models used in estimation of EE on the test datasets. From the table, it can be seen that all the models had similar performance when used in estimating the EE. The R^2 values in Table 4.1 explains the

	RMSE	R-squared
flat	1.576	0.382
SMV_flat	1.487	0.474
equalSegment	1.518	0.430
SMV_equalSegment	1.464	0.467

TABLE 4.1: RMSE and R-square prediction values for the different EE models.

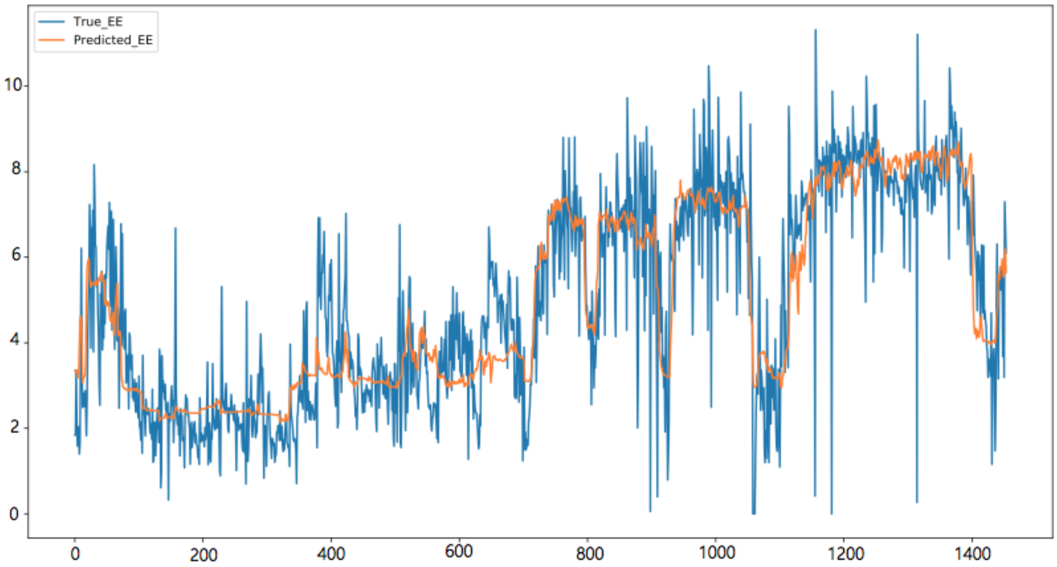


FIGURE 4.12: Energy expenditure graph of the *flat* model.

percentage of variation of the input sensor measurements used in estimating the EE. The higher R^2 value, the better the model when used in estimation of EE [43]. Comparing the R^2 values, the different models can be arranged in increasing order of predictability, namely, *flat*, *equalSegments*, *SMV_equalSegments* and *SMV_flat*. The *SMV_equalSegments* and *SMV_flat* models estimated the EE with similar and higher R^2 values when compared to the *equalSegments* and *flat* models. This makes them better models when used in estimating the EE. To find a more robust model among the 2 (*SMV_equalSegments* and *SMV_flat*), we computed the RMSE for all the models. From table 4.1, it can be seen that the *SMV_flat* has lower generalization error when compared to the *SMV_equalSegments* model. Therefore, making it a more robust model when used in estimation of EE.

Figures 4.12, 4.13, 4.14 and 4.15 show the graphs of EE estimated by the different machine learning models on the same test patient. The blue bar represents the actual EE values and the yellow bar represents the estimated EE values by the machine learning model.

As it can be seen from the graphs, the EE estimated for the different intensity of activities where similar for all the different models. The models performed worst when used in estimating the EE for low intensity activities such as the different types sitting (*sittingChair, sittingSofa, sittingCouch*),

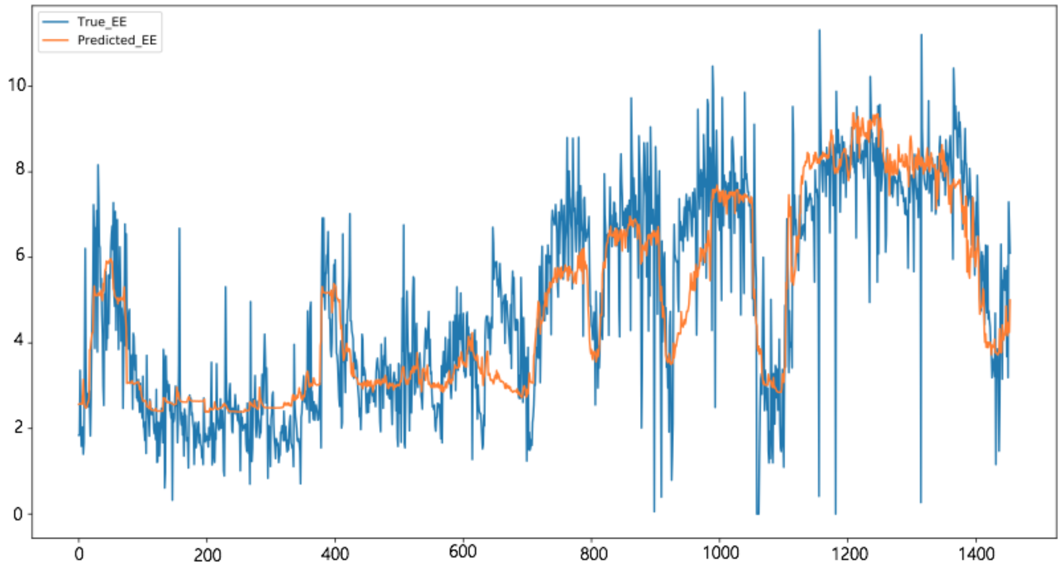


FIGURE 4.13: Energy expenditure graph of the *SMV_flat* model.

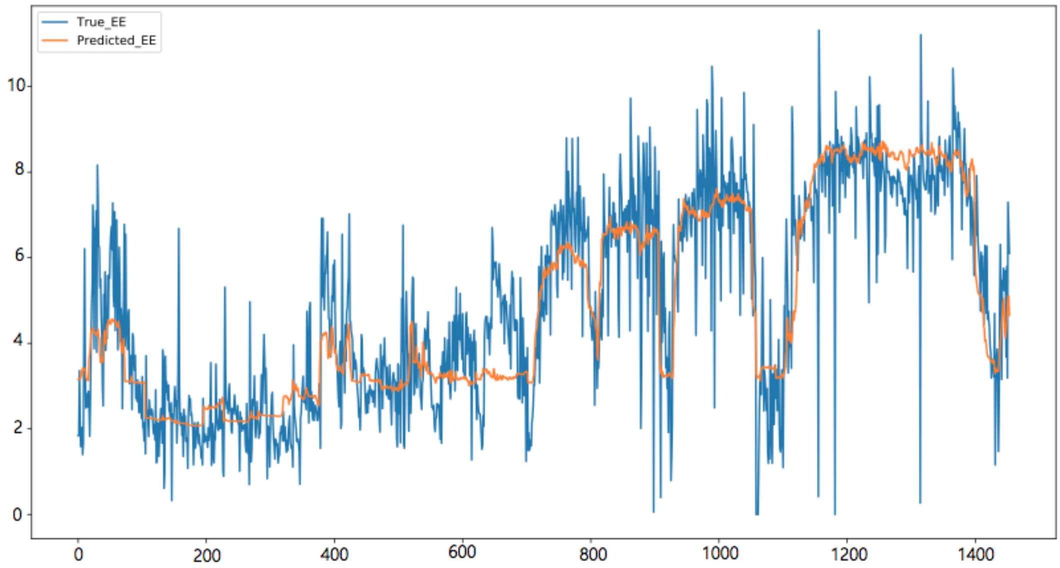


FIGURE 4.14: Energy expenditure graph of the *equalSegment* model.

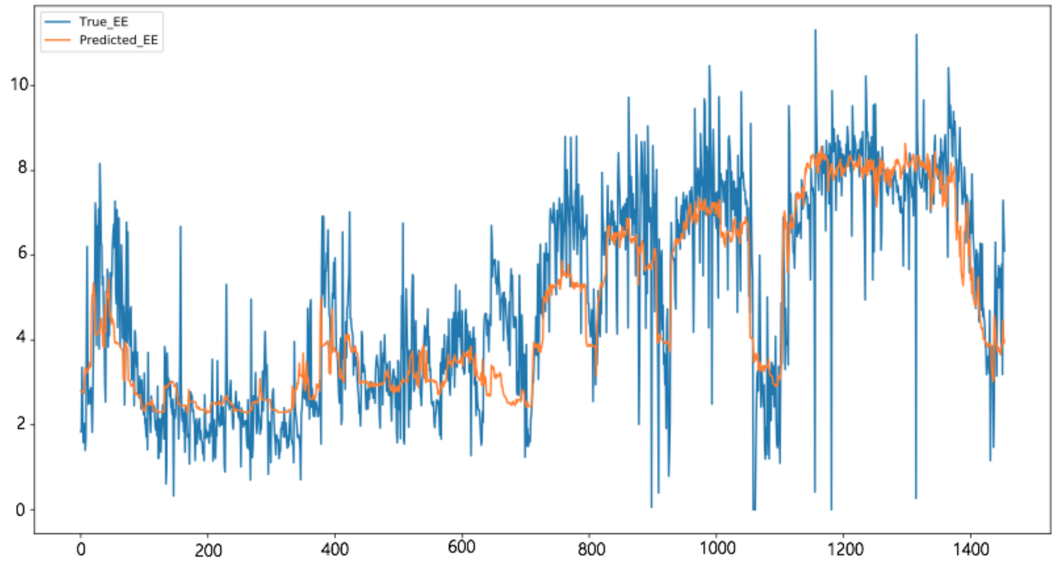


FIGURE 4.15: Energy expenditure graph of the *SMV_equalSegment* model.

lying down (*lyingDownLeft*, *lyingDownRight*) and household activities (*dishwashing*, *stakingShelves*, *vacuumCleaning*). However, the models performed better when used in estimating the EE for high intensity activities such as the different types of walking (*walkingSlow*, *walkingNormal*, *walkingFast*) and *cycling* activities.

Chapter 5

Conclusion

Using sensor and demographic data provided by the Leiden University Medical Center (LUMC), we developed robust and highly accurate deep neural network models for Human Activity Recognition (HAR) and Energy Expenditure (EE). For that, we trained neural network models with Gated Recurrent Units (GRU) and Long Short Term Memory (LSTM) layers. The goal, was to test these models in scenarios where a measurements from one or more sensors could be missing. Therefore, the HAR models had to be robust to missing information and predict the human activities in such situations.

We proposed and tested two neural network approaches for HAR, one using LSTM layers and the other with GRU layers and compared them. The empirical results showed that models with GRU layers performed better when used in HAR. We also compared our GRU models with previous work where Random Forest (RF) models trained specially for each combination of sensor measurements. Also in this case, the empirical results showed that our GRU model was competitive in detecting the human activities against the several specific models of RF.

Based on these experiments, we also concluded that some sensors were more relevant for the detection of the activities studied here. In the case of having two sensors only, the ankle and wrist; or wrist and chest sensors were good at recognizing the activities. This is because most of the activities involved the use of the upper part (arms) or lower part of the body (legs). Therefore, leading to more variation in measurements recorded by these sensors. Also, the wrist or chest sensor is ideal to use for HAR, in the scenario of having one sensor only. However, the chest sensor is better to use, when there is less involvement of the hands or legs in the activities performed.

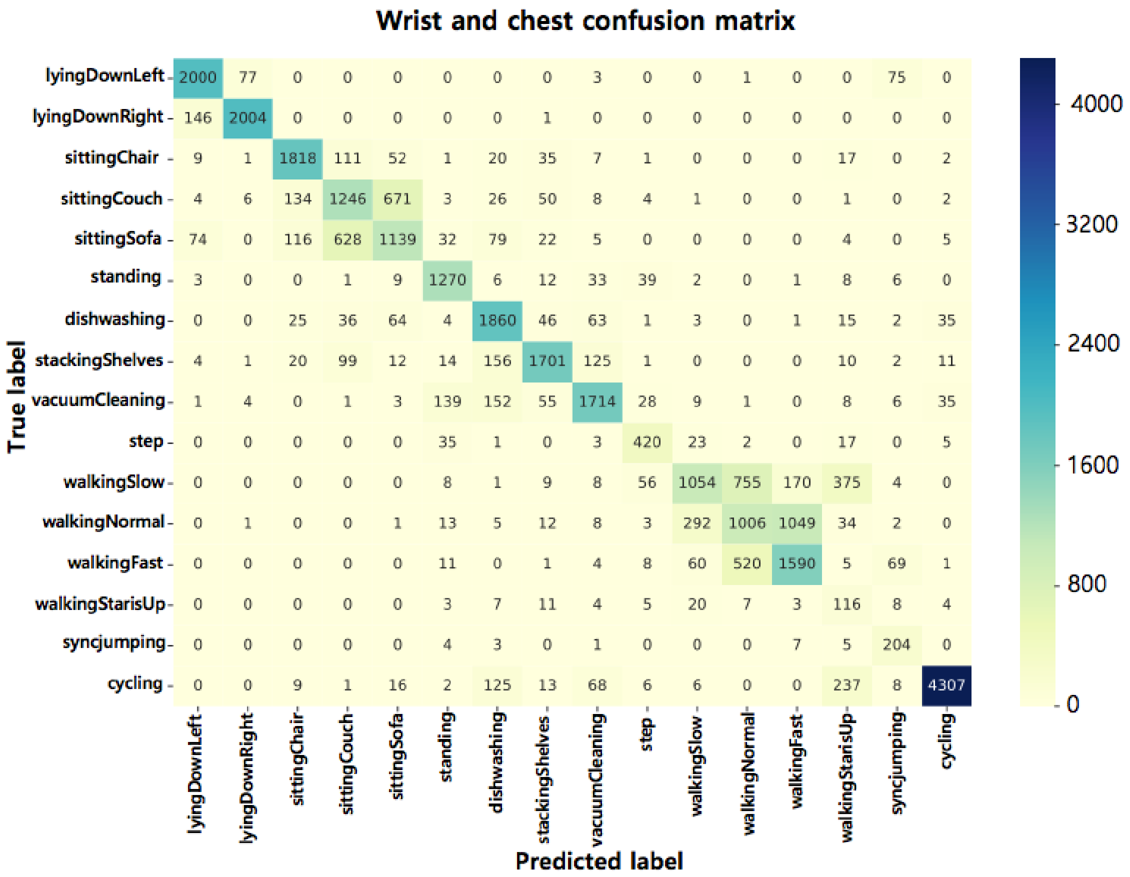
In terms of EE, we used 4 different neural network models for estimating the EE based on different sources of data. Namely, *flat, equalSegments, SVM_equalSegments* and *SVM_flat*. The empirical results showed that all the models estimated the EE with similar performance. These models did not perform so well when estimating the EE for low intensity activities such as the different types sitting (*sittingChair, sittingSofa, sittingCouch*), lying down (*lyingDownLeft, lyingDownRight*) and household activities (*dishwashing, stakingShelves, vacuumCleaning*). However, they performed better when estimating the EE for high intensity activities such as the different types of walking (*walkingSlow, walkingNormal, walkingFast*) and *cycling* activities.

As future work we could investigate the selection of optimal time window that are used for building the sequences for training the machine learning models. Also, further investigation can be done in the tuning of hyper-parameters of the models to improve their performance.

Appendix A

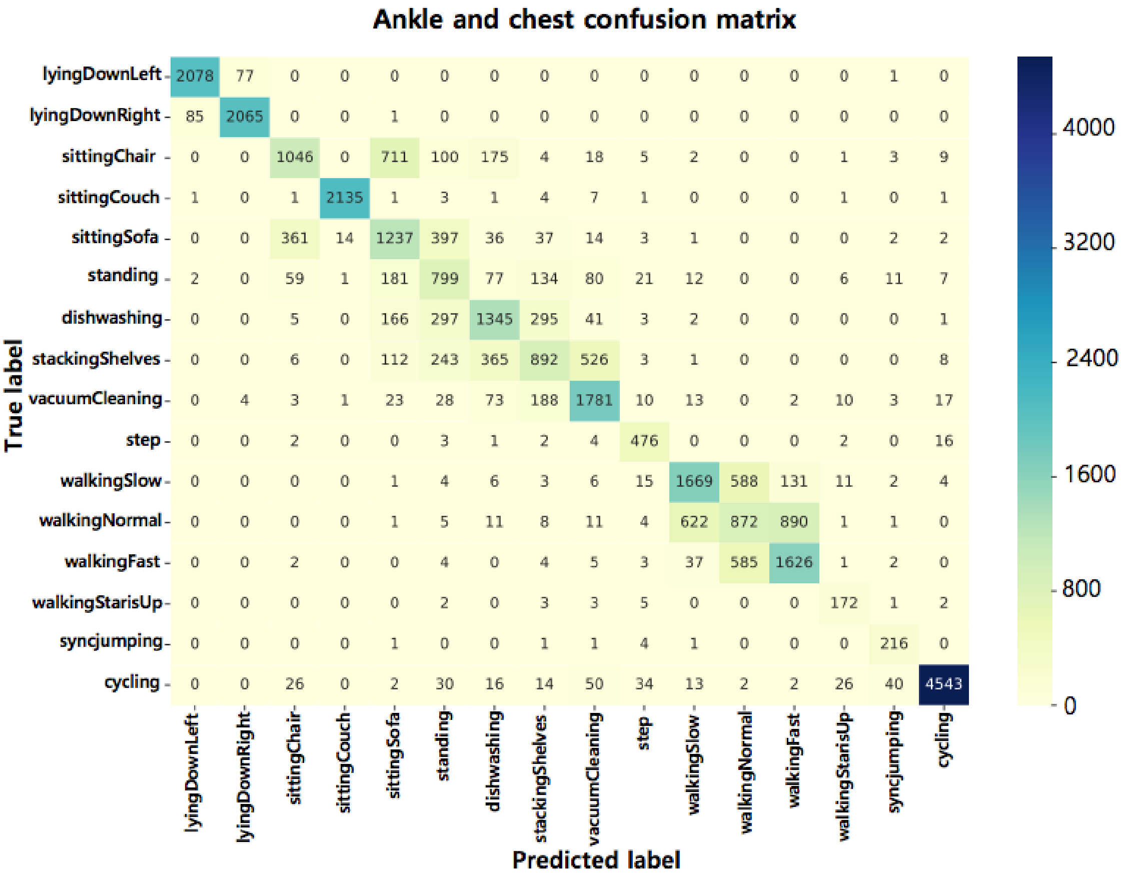
Appendix

A.1 Confusion matrices



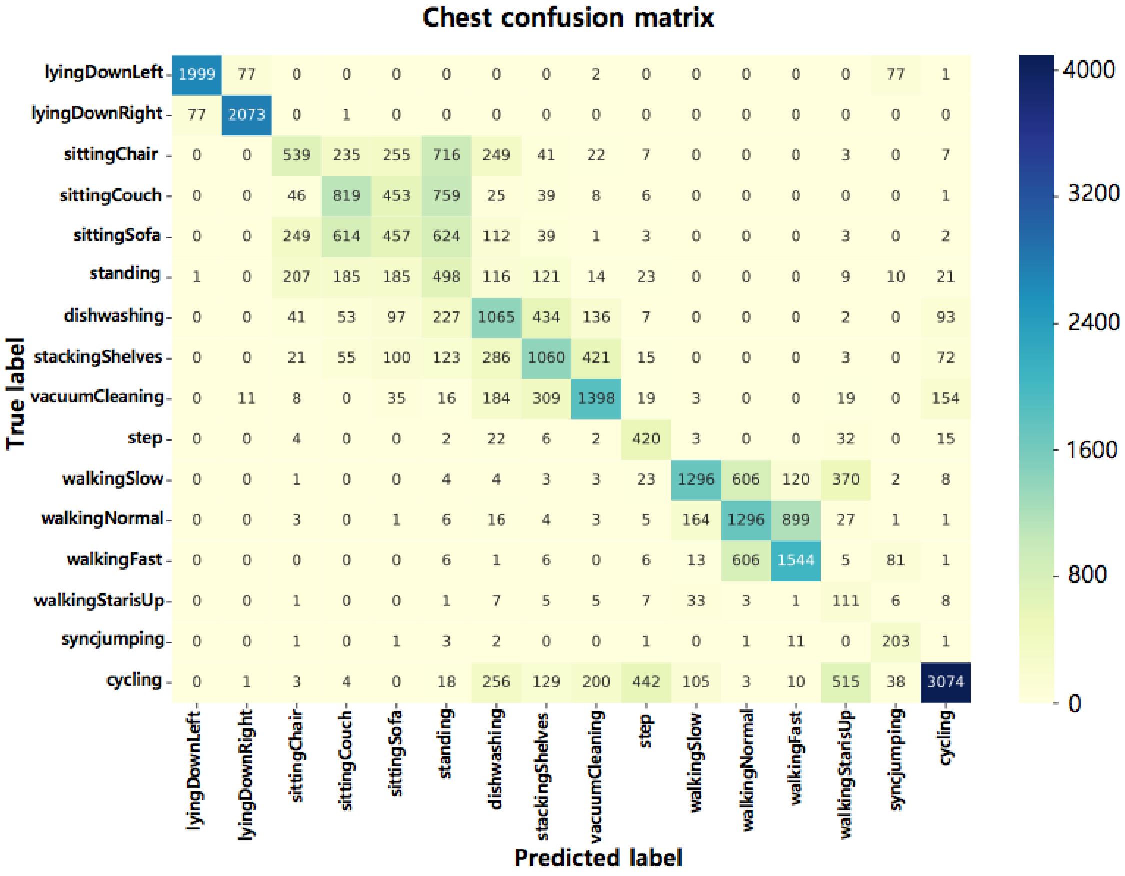
	lyingDownLeft	lyingDownRight	sittingChair	sittingCouch	sittingSofa	standing	dishwashing	stackingShelves	vacuumCleaning	step	walkingSlow	walkingNormal	walkingFast	walkingStarisUp	syncjumping	cycling
lyingDownLeft	2000	77	0	0	0	0	0	0	3	0	0	1	0	0	75	0
lyingDownRight	146	2004	0	0	0	0	0	1	0	0	0	0	0	0	0	0
sittingChair	9	1	1818	111	52	1	20	35	7	1	0	0	0	17	0	2
sittingCouch	4	6	134	1246	671	3	26	50	8	4	1	0	0	1	0	2
sittingSofa	74	0	116	628	1139	32	79	22	5	0	0	0	0	4	0	5
standing	3	0	0	1	9	1270	6	12	33	39	2	0	1	8	6	0
dishwashing	0	0	25	36	64	4	1860	46	63	1	3	0	1	15	2	35
stackingShelves	4	1	20	99	12	14	156	1701	125	1	0	0	0	10	2	11
vacuumCleaning	1	4	0	1	3	139	152	55	1714	28	9	1	0	8	6	35
step	0	0	0	0	0	35	1	0	3	420	23	2	0	17	0	5
walkingSlow	0	0	0	0	0	8	1	9	8	56	1054	755	170	375	4	0
walkingNormal	0	1	0	0	1	13	5	12	8	3	292	1006	1049	34	2	0
walkingFast	0	0	0	0	0	11	0	1	4	8	60	520	1590	5	69	1
walkingStarisUp	0	0	0	0	0	3	7	11	4	5	20	7	3	116	8	4
syncjumping	0	0	0	0	0	4	3	0	1	0	0	0	7	5	204	0
cycling	0	0	9	1	16	2	125	13	68	6	6	0	0	237	8	4307

FIGURE A.1: Confusion matrix of the predictions from all models having wrist and chest sensor measurements.



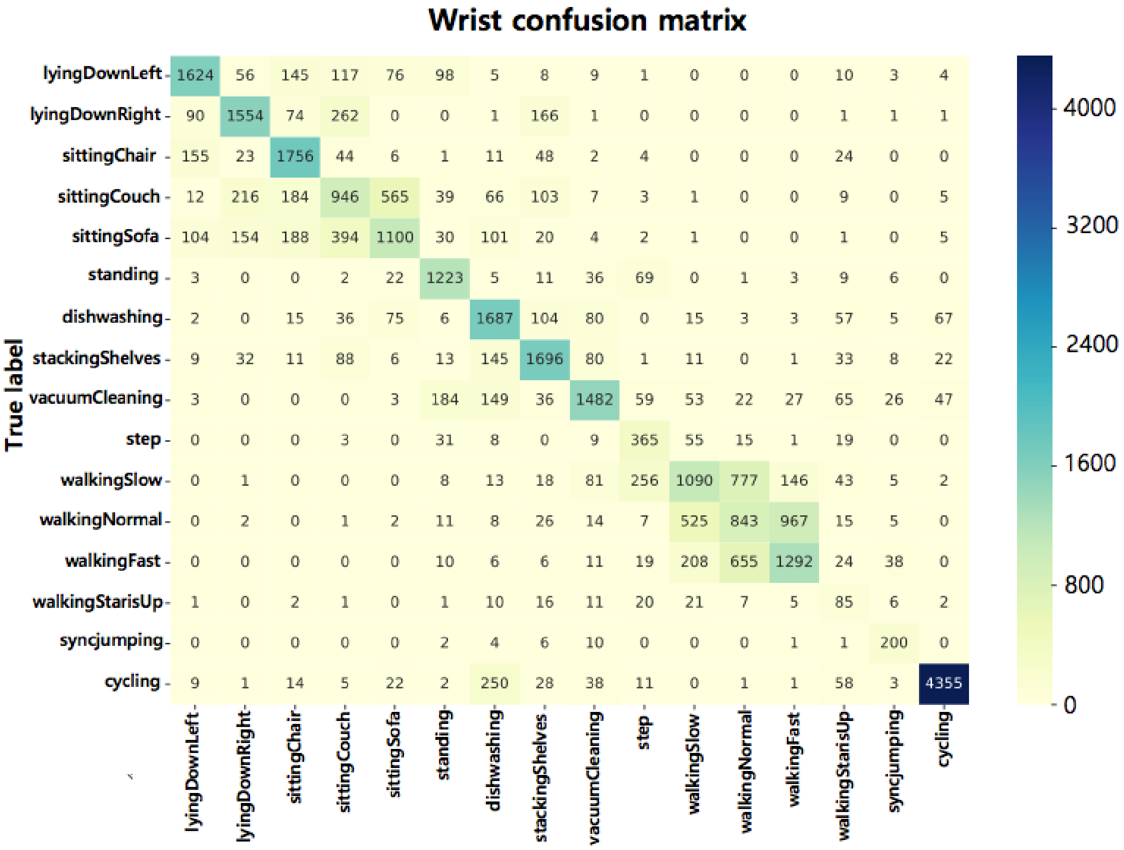
	lyingDownLeft	lyingDownRight	sittingChair	sittingCouch	sittingSofa	standing	dishwashing	stackingShelves	vacuumCleaning	step	walkingSlow	walkingNormal	walkingFast	walkingStarisUp	syncjumping	cycling
lyingDownLeft	2078	77	0	0	0	0	0	0	0	0	0	0	0	0	1	0
lyingDownRight	85	2065	0	0	1	0	0	0	0	0	0	0	0	0	0	0
sittingChair	0	0	1046	0	711	100	175	4	18	5	2	0	0	1	3	9
sittingCouch	1	0	1	2135	1	3	1	4	7	1	0	0	0	1	0	1
sittingSofa	0	0	361	14	1237	397	36	37	14	3	1	0	0	0	2	2
standing	2	0	59	1	181	799	77	134	80	21	12	0	0	6	11	7
dishwashing	0	0	5	0	166	297	1345	295	41	3	2	0	0	0	0	1
stackingShelves	0	0	6	0	112	243	365	892	526	3	1	0	0	0	0	8
vacuumCleaning	0	4	3	1	23	28	73	188	1781	10	13	0	2	10	3	17
step	0	0	2	0	0	3	1	2	4	476	0	0	0	2	0	16
walkingSlow	0	0	0	0	1	4	6	3	6	15	1669	588	131	11	2	4
walkingNormal	0	0	0	0	1	5	11	8	11	4	622	872	890	1	1	0
walkingFast	0	0	2	0	0	4	0	4	5	3	37	585	1626	1	2	0
walkingStarisUp	0	0	0	0	0	2	0	3	3	5	0	0	0	172	1	2
syncjumping	0	0	0	0	1	0	0	1	1	4	1	0	0	0	216	0
cycling	0	0	26	0	2	30	16	14	50	34	13	2	2	26	40	4543

FIGURE A.2: Confusion matrix of the predictions from all models having ankle and chest sensor measurements.



	lyingDownLeft	lyingDownRight	sittingChair	sittingCouch	sittingSofa	standing	dishwashing	stackingShelves	vacuumCleaning	step	walkingSlow	walkingNormal	walkingFast	walkingStarisUp	syncjumping	cycling
lyingDownLeft	1999	77	0	0	0	0	0	0	2	0	0	0	0	0	77	1
lyingDownRight	77	2073	0	1	0	0	0	0	0	0	0	0	0	0	0	0
sittingChair	0	0	539	235	255	716	249	41	22	7	0	0	0	3	0	7
sittingCouch	0	0	46	819	453	759	25	39	8	6	0	0	0	0	0	1
sittingSofa	0	0	249	614	457	624	112	39	1	3	0	0	0	3	0	2
standing	1	0	207	185	185	498	116	121	14	23	0	0	0	9	10	21
dishwashing	0	0	41	53	97	227	1065	434	136	7	0	0	0	2	0	93
stackingShelves	0	0	21	55	100	123	286	1060	421	15	0	0	0	3	0	72
vacuumCleaning	0	11	8	0	35	16	184	309	1398	19	3	0	0	19	0	154
step	0	0	4	0	0	2	22	6	2	420	3	0	0	32	0	15
walkingSlow	0	0	1	0	0	4	4	3	3	23	1296	606	120	370	2	8
walkingNormal	0	0	3	0	1	6	16	4	3	5	164	1296	899	27	1	1
walkingFast	0	0	0	0	0	6	1	6	0	6	13	606	1544	5	81	1
walkingStarisUp	0	0	1	0	0	1	7	5	5	7	33	3	1	111	6	8
syncjumping	0	0	1	0	1	3	2	0	0	1	0	1	11	0	203	1
cycling	0	1	3	4	0	18	256	129	200	442	105	3	10	515	38	3074

FIGURE A.3: Confusion matrix of the predictions from all models having chest sensor measurements only.



	lyingDownLeft	lyingDownRight	sittingChair	sittingCouch	sittingSofa	standing	dishwashing	stackingShelves	vacuumCleaning	step	walkingSlow	walkingNormal	walkingFast	walkingStarisUp	syncjumping	cycling
lyingDownLeft	1624	56	145	117	76	98	5	8	9	1	0	0	0	10	3	4
lyingDownRight	90	1554	74	262	0	0	1	166	1	0	0	0	0	1	1	1
sittingChair	155	23	1756	44	6	1	11	48	2	4	0	0	0	24	0	0
sittingCouch	12	216	184	946	565	39	66	103	7	3	1	0	0	9	0	5
sittingSofa	104	154	188	394	1100	30	101	20	4	2	1	0	0	1	0	5
standing	3	0	0	2	22	1223	5	11	36	69	0	1	3	9	6	0
dishwashing	2	0	15	36	75	6	1687	104	80	0	15	3	3	57	5	67
stackingShelves	9	32	11	88	6	13	145	1696	80	1	11	0	1	33	8	22
vacuumCleaning	3	0	0	0	3	184	149	36	1482	59	53	22	27	65	26	47
step	0	0	0	3	0	31	8	0	9	365	55	15	1	19	0	0
walkingSlow	0	1	0	0	0	8	13	18	81	256	1090	777	146	43	5	2
walkingNormal	0	2	0	1	2	11	8	26	14	7	525	843	967	15	5	0
walkingFast	0	0	0	0	0	10	6	6	11	19	208	655	1292	24	38	0
walkingStarisUp	1	0	2	1	0	1	10	16	11	20	21	7	5	85	6	2
syncjumping	0	0	0	0	0	2	4	6	10	0	0	0	1	1	200	0
cycling	9	1	14	5	22	2	250	28	38	11	0	1	1	58	3	4355

FIGURE A.4: Confusion matrix of the predictions from all models having wrist sensor measurements.

Bibliography

[1] Wil M. P. van der Aalst et al. "Process mining: a two-step approach to balance between underfitting and overfitting". In: *Software and System Modeling* 9.1 (2010), pp. 87–111. DOI: 10.1007/s10270-008-0106-z. URL: https://doi.org/10.1007/s10270-008-0106-z.

[2] Hervé Abdi and Lynne J. Williams. "Using Diode Lasers for Atomic Physics". In: *Wiley Interdisciplinary Reviews: Computational Statistics* 62.1 (Jan. 1991), pp. 1–10. URL: http://link.aip.org/link/?RSI/62/1/1.

[3] Rehan Akbani, Stephen Kwek, and Nathalie Japkowicz. "Applying Support Vector Machines to Imbalanced Datasets". In: *Machine Learning: ECML 2004, 15th European Conference on Machine Learning, Pisa, Italy, September 20-24, 2004, Proceedings*. 2004, pp. 39–50. DOI: 10.1007/978-3-540-30115-8_7. URL: https://doi.org/10.1007/978-3-540-30115-8_7.

[4] Nabil Alshurafa et al. "Designing a Robust Activity Recognition Framework for Health and Exergaming Using Wearable Sensors". In: *IEEE J. Biomedical and Health Informatics* 18.5 (2014), pp. 1636–1646. DOI: 10.1109/JBHI.2013.2287504. URL: https://doi.org/10.1109/JBHI.2013.2287504.

[5] Davide Anguita et al. "Human Activity Recognition on Smartphones Using a Multiclass Hardware-Friendly Support Vector Machine". In: *Ambient Assisted Living and Home Care - 4th International Workshop, IWAAL 2012, Vitoria-Gasteiz, Spain, December 3-5, 2012. Proceedings*. 2012, pp. 216–223. DOI: 10.1007/978-3-642-35395-6_30. URL: https://doi.org/10.1007/978-3-642-35395-6_30.

[6] Ong Chin Ann and Lau Bee Theng. "Human activity recognition: a review". In: *Control System, Computing and Engineering (ICCSCE), 2014 IEEE International Conference on*. IEEE. 2014, pp. 389–393.

[7] Carla Maria Avesani et al. "Physical activity and energy expenditure in haemodialysis patients: an international survey". In: *Nephrology Dialysis Transplantation* 27.6 (2011), pp. 2430–2434.

[8] Michael A Babyak. "What you see may not be what you get: a brief, nontechnical introduction to overfitting in regression-type models". In: *Psychosomatic medicine* 66.3 (2004), pp. 411–421.

[9] Oresti Baños et al. "Daily living activity recognition based on statistical feature quality group selection". In: *Expert Syst. Appl.* 39.9 (2012), pp. 8013–8021. DOI: 10.1016/j.eswa.2012.01.164. URL: https://doi.org/10.1016/j.eswa.2012.01.164.

[10] Mathieu Barnachon et al. “A real-time system for motion retrieval and interpretation”. In: *Pattern Recognition Letters* 34.15 (2013), pp. 1789–1798. DOI: 10.1016/j.patrec.2012.12.020. URL: https://doi.org/10.1016/j.patrec.2012.12.020.

[11] Akram Bayat, Marc Pomplun, and Duc A. Tran. “A Study on Human Activity Recognition Using Accelerometer Data from Smartphones”. In: *The 9th International Conference on Future Networks and Communications (FNC'14) / The 11th International Conference on Mobile Systems and Pervasive Computing (MobiSPC'14) / Affiliated Workshops, August 17-20, 2014, Niagara Falls, Canada*. 2014, pp. 450–457. DOI: 10.1016/j.procs.2014.07.009. URL: https://doi.org/10.1016/j.procs.2014.07.009.

[12] Yoshua Bengio, Patrice Y. Simard, and Paolo Frasconi. “Learning long-term dependencies with gradient descent is difficult”. In: *IEEE Trans. Neural Networks* 5.2 (1994), pp. 157–166. DOI: 10.1109/72.279181. URL: https://doi.org/10.1109/72.279181.

[13] Sebastian D. Bersch et al. “Sensor Data Acquisition and Processing Parameters for Human Activity Classification”. In: *Sensors* 14.3 (2014), pp. 4239–4270. DOI: 10.3390/s140304239. URL: https://doi.org/10.3390/s140304239.

[14] Leo Breiman. “Random forests”. In: *Machine learning* 45.1 (2001), pp. 5–32.

[15] Ricardo Cachucho et al. “Mining Multivariate Time Series with Mixed Sampling Rates”. In: *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. UbiComp '14. Seattle, Washington: ACM, 2014, pp. 413–423. ISBN: 978-1-4503-2968-2. DOI: 10.1145/2632048.2632061. URL: http://doi.acm.org/10.1145/2632048.2632061.

[16] Pierluigi Casale, Oriol Pujol, and Petia Radeva. “Human Activity Recognition from Accelerometer Data Using a Wearable Device”. In: *Pattern Recognition and Image Analysis - 5th Iberian Conference, IbPRIA 2011, Las Palmas de Gran Canaria, Spain, June 8-10, 2011. Proceedings*. 2011, pp. 289–296. DOI: 10.1007/978-3-642-21257-4_36. URL: https://doi.org/10.1007/978-3-642-21257-4_36.

[17] Alexandros André Chaaraoui et al. “Evolutionary joint selection to improve human action recognition with RGB-D devices”. In: *Expert Syst. Appl.* 41.3 (2014), pp. 786–794. DOI: 10.1016/j.eswa.2013.08.009. URL: https://doi.org/10.1016/j.eswa.2013.08.009.

[18] Haowen Chan and Adrian Perrig. “Security and privacy in sensor networks”. In: *computer* 36.10 (2003), pp. 103–105.

[19] Nitesh V Chawla. “C4. 5 and imbalanced data sets: investigating the effect of sampling method, probabilistic estimate, and decision tree structure”. In: *Proceedings of the ICML*. Vol. 3. 2003, p. 66.

[20] Liming Chen, Chris D. Nugent, and Hui Wang. "A Knowledge-Driven Approach to Activity Recognition in Smart Homes". In: *IEEE Trans. Knowl. Data Eng.* 24.6 (2012), pp. 961–974. DOI: 10.1109/TKDE.2011.51. URL: https://doi.org/10.1109/TKDE.2011.51.

[21] Lulu Chen, Hong Wei, and James M. Ferryman. "A survey of human motion analysis using depth imagery". In: *Pattern Recognition Letters* 34.15 (2013), pp. 1995–2006. DOI: 10.1016/j.patrec.2013.02.006. URL: https://doi.org/10.1016/j.patrec.2013.02.006.

[22] Junyoung Chung et al. "Gated Feedback Recurrent Neural Networks". In: *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*. 2015, pp. 2067–2075. URL: http://jmlr.org/proceedings/papers/v37/chung15.html.

[23] Bo Dong et al. "Comparing metabolic energy expenditure estimation using wearable multi-sensor network and single accelerometer". In: *35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC 2013, Osaka, Japan, July 3-7, 2013*. 2013, pp. 2866–2869. DOI: 10.1109/EMBC.2013.6610138. URL: https://doi.org/10.1109/EMBC.2013.6610138.

[24] Yong Du, Wei Wang, and Liang Wang. "Hierarchical recurrent neural network for skeleton based action recognition". In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*. 2015, pp. 1110–1118. DOI: 10.1109/CVPR.2015.7298714. URL: https://doi.org/10.1109/CVPR.2015.7298714.

[25] Thi V. Duong et al. "Activity Recognition and Abnormality Detection with the Switching Hidden Semi-Markov Model". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*. 2005, pp. 838–845. DOI: 10.1109/CVPR.2005.61. URL: https://doi.org/10.1109/CVPR.2005.61.

[26] Sean R. Eddy. "Profile hidden Markov models." In: *Bioinformatics (Oxford, England)* 14.9 (1998), pp. 755–763.

[27] Katherine Ellis et al. "A random forest classifier for the prediction of energy expenditure and type of physical activity from wrist and hip accelerometers". In: *Physiological measurement* 35.11 (2014), p. 2191.

[28] Michael Esterman et al. "Avoiding non-independence in fMRI data analysis: Leave one subject out". In: *NeuroImage* 50.2 (2010), pp. 572–576. DOI: 10.1016/j.neuroimage.2009.10.092. URL: https://doi.org/10.1016/j.neuroimage.2009.10.092.

[29] Alberto Fernández, Salvador García, and Francisco Herrera. "Addressing the Classification with Imbalanced Data: Open Problems and New Challenges on Class Distribution". In: *Hybrid Artificial Intelligent Systems - 6th International Conference, HAIS 2011, Wroclaw, Poland, May 23-25, 2011, Proceedings, Part I*. 2011, pp. 1–10. DOI: 10.1007/978-3-642-21219-2_1. URL: https://doi.org/10.1007/978-3-642-21219-2_1.

[30] Ronald Fischer. “Standardization to account for cross-cultural response bias: A classification of score adjustment procedures and review of research in JCCP”. In: *Journal of Cross-Cultural Psychology* 35.3 (2004), pp. 263–282.

[31] Matt W Gardner and SR Dorling. “Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences”. In: *Atmospheric environment* 32.14-15 (1998), pp. 2627–2636.

[32] Stanton A Glantz, Bryan K Slinker, and Torsten B Neilands. *Primer of applied regression and analysis of variance*. Vol. 309. McGraw-Hill New York, 1990.

[33] Anjela Govan. “Introduction to optimization”. In: *North Carolina State University, SAMSI NDHS, Undergraduate workshop*. 2006.

[34] Alex Graves. *Supervised Sequence Labelling with Recurrent Neural Networks*. Vol. 385. Studies in Computational Intelligence. Springer, 2012. ISBN: 978-3-642-24796-5. DOI: 10.1007/978-3-642-24797-2. URL: https://doi.org/10.1007/978-3-642-24797-2.

[35] Isabelle Guyon and André Elisseeff. “An introduction to feature extraction”. In: *Feature extraction*. Springer, 2006, pp. 1–25.

[36] Isabelle Guyon and André Elisseeff. “An introduction to variable and feature selection”. In: *Journal of machine learning research* 3.Mar (2003), pp. 1157–1182.

[37] Martin T. Hagan and Mohammad B. Menhaj. “Training feedforward networks with the Marquardt algorithm”. In: *IEEE Trans. Neural Networks* 5.6 (1994), pp. 989–993. DOI: 10.1109/72.329697. URL: https://doi.org/10.1109/72.329697.

[38] Marc T Hamilton, Deborah G Hamilton, and Theodore W Zderic. “The role of low energy expenditure and sitting on obesity, metabolic syndrome, type 2 diabetes, and cardiovascular disease”. In: *Diabetes* (2007).

[39] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. “Unsupervised learning”. In: *The elements of statistical learning*. Springer, 2009, pp. 485–585.

[40] Sepp Hochreiter and Jürgen Schmidhuber. “Long Short-Term Memory”. In: *Neural Computation* 9.8 (1997), pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735. URL: https://doi.org/10.1162/neco.1997.9.8.1735.

[41] Yuhuang Hu et al. “Overcoming the vanishing gradient problem in plain recurrent networks”. In: *CoRR* abs/1801.06105 (2018). arXiv: 1801.06105. URL: http://arxiv.org/abs/1801.06105.

[42] Jun-Da Huang. “Kinerehab: a kinect-based system for physical rehabilitation: a pilot study for young adults with motor disabilities”. In: *The 13th International ACM SIGACCESS Conference on Computers and Accessibility, ASSETS '11, Dundee, Scotland, UK, October 24-26, 2011*. 2011, pp. 319–320. DOI: 10.1145/2049536.2049627. URL: https://doi.org/10.1145/2049536.2049627.

[43] Rob J Hyndman and Anne B Koehler. "Another look at measures of forecast accuracy". In: *International journal of forecasting* 22.4 (2006), pp. 679–688.

[44] Cangzhi Jia, Yun Zuo, and Quan Zou. "O-GlcNAcPRED-II: an integrated classification algorithm for identifying O-GlcNAcylation sites based on fuzzy undersampling and a K-means PCA oversampling technique". In: *Bioinformatics* 34.12 (2018), pp. 2029–2036. DOI: 10.1093/bioinformatics/bty039. URL: https://doi.org/10.1093/bioinformatics/bty039.

[45] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. "Reinforcement learning: A survey". In: *Journal of artificial intelligence research* 4 (1996), pp. 237–285.

[46] Adil Mehmood Khan et al. "A triaxial accelerometer-based physical-activity recognition via augmented-signal features and a hierarchical recognizer". In: *IEEE Trans. Information Technology in Biomedicine* 14.5 (2010), pp. 1166–1172. DOI: 10.1109/TITB.2010.2051955. URL: https://doi.org/10.1109/TITB.2010.2051955.

[47] Eunju Kim, Sumi Helal, and Diane J. Cook. "Human Activity Recognition and Pattern Discovery". In: *IEEE Pervasive Computing* 9.1 (2010), pp. 48–53. DOI: 10.1109/MPRV.2010.7. URL: https://doi.org/10.1109/MPRV.2010.7.

[48] Stefan Knerr, Léon Personnaz, and Gérard Dreyfus. "Single-layer learning revisited: a stepwise procedure for building and training a neural network". In: *Neurocomputing*. Springer, 1990, pp. 41–50.

[49] Sotiris B Kotsiantis, I Zaharakis, and P Pintelas. "Supervised machine learning: A review of classification techniques". In: *Emerging artificial intelligence applications in computer engineering* 160 (2007), pp. 3–24.

[50] John R Koza et al. "Automated design of both the topology and sizing of analog electrical circuits using genetic programming". In: *Artificial Intelligence in Design'96*. Springer, 1996, pp. 151–170.

[51] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States.* 2012, pp. 1106–1114. URL: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.

[52] Walter S. Lasecki et al. "Real-time crowd labeling for deployable activity recognition". In: *Computer Supported Cooperative Work, CSCW 2013, San Antonio, TX, USA, February 23-27, 2013*. 2013, pp. 1203–1212. DOI: 10.1145/2441776.2441912. URL: https://doi.org/10.1145/2441776.2441912.

[53] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *nature* 521.7553 (2015), p. 436.

[54] Kyungmin Lee et al. "Accelerating Recurrent Neural Network Language Model Based Online Speech Recognition System". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*. 2018, pp. 5904–5908. DOI: 10.1109/ICASSP.2018.8462334. URL: https://doi.org/10.1109/ICASSP.2018.8462334.

[55] Camelia Lemnaru and Rodica Potolea. "Imbalanced Classification Problems: Systematic Study, Issues and Best Practices". In: *Enterprise Information Systems - 13th International Conference, ICEIS 2011, Beijing, China, June 8-11, 2011, Revised Selected Papers*. 2011, pp. 35–50. DOI: 10.1007/978-3-642-29958-2_3. URL: https://doi.org/10.1007/978-3-642-29958-2_3.

[56] Shuaimin Li and Jungang Xu. "A Recurrent Neural Network Language Model Based on Word Embedding". In: *Web and Big Data - APWeb-WAIM 2018 International Workshops: MWDA, BAH, KGMA, DMMOOC, DS, Macau, China, July 23-25, 2018, Revised Selected Papers*. 2018, pp. 368–377. DOI: 10.1007/978-3-030-01298-4_30. URL: https://doi.org/10.1007/978-3-030-01298-4_30.

[57] Richard Lippmann. "An introduction to computing with neural nets". In: *IEEE Assp magazine* 4.2 (1987), pp. 4–22.

[58] Victoria López et al. "Analysis of preprocessing vs. cost-sensitive learning for imbalanced classification. Open problems on intrinsic data characteristics". In: *Expert Syst. Appl.* 39.7 (2012), pp. 6585–6608. DOI: 10.1016/j.eswa.2011.12.043. URL: https://doi.org/10.1016/j.eswa.2011.12.043.

[59] Todd M Manini et al. "Daily activity energy expenditure and mortality among older adults". In: *Jama* 296.2 (2006), pp. 171–179.

[60] Andrea Mannini and Angelo Maria Sabatini. "Machine Learning Methods for Classifying Human Physical Activity from On-Body Accelerometers". In: *Sensors* 10.2 (2010), pp. 1154–1175. DOI: 10.3390/s100201154. URL: https://doi.org/10.3390/s100201154.

[61] Tomohiro Mashita et al. "Human activity recognition for a content search system considering situations of smartphone users". In: *Virtual Reality Short Papers and Posters (VRW), 2012 IEEE*. IEEE. 2012, pp. 1–2.

[62] Luis Carlos Molina, Lluís Belanche, and Àngela Nebot. "Feature selection algorithms: A survey and experimental evaluation". In: *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*. IEEE. 2002, pp. 306–313.

[63] David J Montana and Lawrence Davis. "Training Feedforward Neural Networks Using Genetic Algorithms." In: *IJCAI*. Vol. 89. 1989, pp. 762–767.

[64] Alexander HK Montoye et al. "Comparison of linear and non-linear models for predicting energy expenditure from raw accelerometer data". In: *Physiological measurement* 38.2 (2017), p. 343.

[65] Alexander HK Montoye et al. "Wrist-independent energy expenditure prediction models from raw accelerometer data". In: *Physiological measurement* 37.10 (2016), p. 1770.

[66] Raymond H Myers and Raymond H Myers. *Classical and modern regression with applications*. Vol. 2. Duxbury press Belmont, CA, 1990.

[67] Andrew Y Ng. "Feature selection, L 1 vs. L 2 regularization, and rotational invariance". In: *Proceedings of the twenty-first international conference on Machine learning*. ACM. 2004, p. 78.

[68] Nattapon Noorit and Nikom Suvonvorn. "Human Activity Recognition from Basic Actions Using Finite State Machine". In: *Proceedings of the First International Conference on Advanced Data and Information Engineering, DaEng 2013, Kuala Lumpur, Malaysia, December 16-18, 2013.* 2013, pp. 379–386. DOI: 10.1007/978-981-4585-18-7_43. URL: https://doi.org/10.1007/978-981-4585-18-7_43.

[69] Wee-Hong Ong, Leon Palafox, and Takafumi Koseki. "Investigation of feature extraction for unsupervised learning in human activity detection". In: *Bulletin of Networking, Computing, Systems, and Software* 2.1 (2013), pp–30.

[70] Kazuki Osawa et al. "Second-order Optimization Method for Large Mini-batch: Training ResNet-50 on ImageNet in 35 Epochs". In: *CoRR* abs/1811.12019 (2018). arXiv: 1811.12019. URL: http://arxiv.org/abs/1811.12019.

[71] Stylianos Paraschiakos. "Comparing Sensor Networks for Activity Recognition". In: *Comparing Sensor Networks for Activity Recognition* 1 (Jan. 2017), pp. 1–69.

[72] Yubin Park and Joydeep Ghosh. "Ensembles of (α)-Trees for Imbalanced Classification Problems". In: *IEEE Trans. Knowl. Data Eng.* 26.1 (2014), pp. 131–143. DOI: 10.1109/TKDE.2012.255. URL: https://doi.org/10.1109/TKDE.2012.255.

[73] Tina R Patil and SS Sherekar. "Performance analysis of Naive Bayes and J48 classification algorithm for data classification". In: *International journal of computer science and applications* 6.2 (2013), pp. 256–261.

[74] William Pavot et al. "Further validation of the Satisfaction with Life Scale: Evidence for the cross-method convergence of well-being measures". In: *Journal of personality assessment* 57.1 (1991), pp. 149–161.

[75] Luis Perez and Jason Wang. "The Effectiveness of Data Augmentation in Image Classification using Deep Learning". In: *CoRR* abs/1712.04621 (2017). arXiv: 1712.04621. URL: http://arxiv.org/abs/1712.04621.

[76] Guy Plasqui and Klaas R Westerterp. "Physical activity assessment with accelerometers: an evaluation against doubly labeled water". In: *Obesity* 15.10 (2007), pp. 2371–2379.

[77] Lawrence R Rabiner. “A tutorial on hidden Markov models and selected applications in speech recognition”. In: *Proceedings of the IEEE* 77.2 (1989), pp. 257–286.

[78] Lawrence R Rabiner and Bernard Gold. “Theory and application of digital signal processing”. In: *Englewood Cliffs, NJ, Prentice-Hall, Inc., 1975. 777 p.* (1975).

[79] Nishkam Ravi et al. “Activity recognition from accelerometer data”. In: *Aaai*. Vol. 5. 2005. 2005, pp. 1541–1546.

[80] Christian Robert. *Machine learning, a probabilistic perspective*. 2014.

[81] Mehrsan Javan Roshtkhari and Martin D. Levine. “Human activity recognition in videos using a single example”. In: *Image Vision Comput.* 31.11 (2013), pp. 864–876. DOI: 10.1016/j.imavis.2013.08.005. URL: https://doi.org/10.1016/j.imavis.2013.08.005.

[82] Megan P Rothney et al. “An artificial neural network model of energy expenditure using nonintegrated acceleration signals”. In: *Journal of applied physiology* 103.4 (2007), pp. 1419–1427.

[83] Michael S. Ryoo. “Human activity prediction: Early recognition of ongoing activities from streaming videos”. In: *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011*. 2011, pp. 1036–1043. DOI: 10.1109/ICCV.2011.6126349. URL: https://doi.org/10.1109/ICCV.2011.6126349.

[84] Maryam A. Salih et al. “Hybrid module for low-cost surveillance system”. In: *6th International Conference on Information and Communication Technology and Accessibility, ICTA 2017, Muscat, Oman, December 19-21, 2017*. 2017, pp. 1–7. DOI: 10.1109/ICTA.2017.8336012. URL: https://doi.org/10.1109/ICTA.2017.8336012.

[85] Edward Sazonov et al. “Monitoring of Posture Allocations and Activities by a Shoe-Based Wearable Sensor”. In: *IEEE Trans. Biomed. Engineering* 58.4 (2011), pp. 983–990. DOI: 10.1109/TBME.2010.2046738. URL: https://doi.org/10.1109/TBME.2010.2046738.

[86] Arvind Kumar Shekar et al. “Building robust prediction models for defective sensor data using Artificial Neural Networks”. In: *CoRR* abs/1804.05544 (2018). arXiv: 1804.05544. URL: http://arxiv.org/abs/1804.05544.

[87] Patrice Y. Simard, David Steinkraus, and John C. Platt. “Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis”. In: *7th International Conference on Document Analysis and Recognition (ICDAR 2003), 2-Volume Set, 3-6 August 2003, Edinburgh, Scotland, UK*. 2003, pp. 958–962. DOI: 10.1109/ICDAR.2003.1227801. URL: https://doi.org/10.1109/ICDAR.2003.1227801.

[88] Pawan Kumar Singh, Ram Sarkar, and Mita Nasipuri. “Statistical validation of multiple classifiers over multiple datasets in the field of pattern recognition”. In: *IJAPR* 2.1 (2015), pp. 1–23. DOI: 10.1504/IJAPR.2015.068929. URL: https://doi.org/10.1504/IJAPR.2015.068929.

[89] Nitish Srivastava et al. “Dropout: a simple way to prevent neural networks from overfitting”. In: *Journal of Machine Learning Research* 15.1 (2014), pp. 1929–1958. URL: http://dl.acm.org/citation.cfm?id=2670313.

[90] D Stathakis. “How many hidden layers and nodes?” In: *International Journal of Remote Sensing* 30.8 (2009), pp. 2133–2147.

[91] John Staudenmayer et al. “An artificial neural network to estimate physical activity energy expenditure and identify physical activity type from an accelerometer”. In: *Journal of Applied Physiology* 107.4 (2009), pp. 1300–1307.

[92] José A. Sáez et al. “SMOTE–IPF: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering”. In: *Information Sciences* 291 (2015), pp. 184 – 203. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2014.08.051. URL: http://www.sciencedirect.com/science/article/pii/S0020025514008561.

[93] Masaki Takahashi et al. “Robust Recognition of Specific Human Behaviors in Crowded Surveillance Video Sequences”. In: *EURASIP J. Adv. Sig. Proc.* 2010 (2010). DOI: 10.1155/2010/801252. URL: https://doi.org/10.1155/2010/801252.

[94] Martin Thoma. “Analysis and Optimization of Convolutional Neural Network Architectures”. In: *CoRR* abs/1707.09725 (2017). arXiv: 1707.09725. URL: http://arxiv.org/abs/1707.09725.

[95] David A Van Dyk and Xiao-Li Meng. “The art of data augmentation”. In: *Journal of Computational and Graphical Statistics* 10.1 (2001), pp. 1–50.

[96] William J Vincent and Joseph P Weir. *Statistics in kinesiology*. Human Kinetics, 2018.

[97] Quang Viet Vo, Gueesang Lee, and Deokjai Choi. “Fall Detection Based on Movement and Smart Phone Technology”. In: *2012 IEEE RIVF International Conference on Computing & Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), Ho Chi Minh City, Vietnam, February 27 - March 1, 2012*. 2012, pp. 1–4. DOI: 10.1109/rivf.2012.6169847. URL: https://doi.org/10.1109/rivf.2012.6169847.

[98] Sérgio B. Volchan. “What Is a Random Sequence?” In: *The American Mathematical Monthly* 109.1 (2002), pp. 46–63. URL: http://www.jstor.org/stable/2695767.

[99] Nisarg Vyas et al. “Machine learning and sensor fusion for estimating continuous energy expenditure”. In: *AI Magazine* 33.2 (2012), p. 55.

[100] Simin Wang, Salim Zabir, and Bastian Leibe. “Lying Pose Recognition for Elderly Fall Detection”. In: *Robotics: Science and Systems VII, University of Southern California, Los Angeles, CA, USA, June 27-30, 2011*. 2011. DOI: 10.15607/RSS.2011.VII.044. URL: http://www.roboticsproceedings.org/rss07/p44.html.

[101] Darren ER Warburton, Crystal Whitney Nicol, and Shannon SD Bredin. “Health benefits of physical activity: the evidence”. In: *Canadian medical association journal* 174.6 (2006), pp. 801–809.

[102] Iris Weller and Paul Corey. “The impact of excluding non-leisure energy expenditure on the relation between physical activity and mortality in women”. In: *Epidemiology* (1998), pp. 632–635.

[103] Jaeyoung Yang, Joonwhan Lee, and Joongmin Choi. “Activity Recognition Based on RFID Object Usage for Smart Mobile Devices”. In: *J. Comput. Sci. Technol.* 26.2 (2011), pp. 239–246. DOI: `10.1007/s11390-011-9430-9`. URL: `https://doi.org/10.1007/s11390-011-9430-9`.

[104] Aref Yelghi and Cemal Köse. “A modified firefly algorithm for global minimum optimization”. In: *Appl. Soft Comput.* 62 (2018), pp. 29–44. DOI: `10.1016/j.asoc.2017.10.032`. URL: `https://doi.org/10.1016/j.asoc.2017.10.032`.