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 Okaï

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

Abstract i
Acknowledgements ii

1 Introduction 1

2 Related Work 3
  2.1 Human Activity Recognition (HAR) 3
  2.2 Energy Expenditure (EE) 3

3 Materials and Methods 5
  3.1 Basic concepts of Machine Learning 5
    3.1.1 Regularization 5
    3.1.2 Evaluation 6
      Learning curves 6
      Confusion matrix 7
      Evaluation Metrics 8
    3.1.3 Gradient descent 8
    3.1.4 Data processing 9
      Resampling 9
      Feature extraction 10
      Feature selection 10
      Downsampling 11
      Standardization 11
  3.2 Artificial Neural Networks 11
    3.2.1 Recurrent Neural Networks 12
      Long short-term memory (LSTM) 12
      Gated recurrent units (GRUs) 13
    3.2.2 Dropout 14
  3.3 Data Augmentation 14
  3.4 Data 15

4 Experimental Setup and Results 20
  4.1 Human Activity Recognition 20
    4.1.1 Data pre-processing 20
    4.1.2 Proposed approaches 21
    Data Generator 23
    4.1.3 Evaluation 23
    4.1.4 GRU approach 25
      Ankle, wrist and chest confusion matrix 27
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ankle and wrist confusion matrix</td>
<td>27</td>
</tr>
<tr>
<td>Ankle confusion matrix</td>
<td>28</td>
</tr>
<tr>
<td>4.1.5 LSTM approach</td>
<td>29</td>
</tr>
<tr>
<td>4.2 Energy Expenditure</td>
<td>30</td>
</tr>
<tr>
<td>4.2.1 Proposed approach</td>
<td>31</td>
</tr>
<tr>
<td>4.2.2 Data pre-processing</td>
<td>34</td>
</tr>
<tr>
<td>Window Size</td>
<td>34</td>
</tr>
<tr>
<td>4.2.3 Evaluation</td>
<td>36</td>
</tr>
<tr>
<td>4.2.4 EE Models</td>
<td>36</td>
</tr>
<tr>
<td>5 Conclusion</td>
<td>40</td>
</tr>
<tr>
<td>A Appendix</td>
<td>42</td>
</tr>
<tr>
<td>A.1 Confusion matrices</td>
<td>42</td>
</tr>
<tr>
<td>Bibliography</td>
<td>46</td>
</tr>
</tbody>
</table>
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.
Chapter 1. Introduction

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.

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 Neighbors (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 inductive 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.
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
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 (FP) – 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.

<table>
<thead>
<tr>
<th>Actual value</th>
<th>p’</th>
<th>n’</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>p’</td>
<td>True Positive</td>
<td>False Negative</td>
<td>p’</td>
</tr>
<tr>
<td>n’</td>
<td>False Positive</td>
<td>True Negative</td>
<td>n’</td>
</tr>
<tr>
<td>total</td>
<td>P</td>
<td>N</td>
<td></td>
</tr>
</tbody>
</table>

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:

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

**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 minum 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].
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
Chapter 3. Materials and Methods

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].
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.
Chapter 3. Materials and Methods

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.
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.
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 measurements) 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.

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

Because of privacy reasons, we do not present the personal codes of the patients.
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, stacking Shelves and vacuum Cleaning. The outdoor activities included the different types of walking walking Slow, walking Normal, walking Fast and cycling. All patients were allowed a resting period, maximum of 1 minute between the different activities performed, by standing.

Even though each patient 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.

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

1: $D =$ Original dataset
2: $s =$ Sequence size
3: $P =$ Patients
4: $Q =$ array of sequences
5: **procedure** MAKESEQUENCES($D, s$)
6:   **for each** patient $p$ in $P$ do
7:     $A \leftarrow$ Select all activities of patient $p$ in $D$
8:     **for each** activity $j$ in $A$ do
9:     Select all sensor measurements from $D$ for patient $p$ and activity $j$
10:     Create sequences of length $s$ and assign the sequences to $Q$
11:   **end for**
12: **end for**
13: **return** $Q$ \(\triangleright\) return sequences for all patients
14: **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
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 4.1: Distribution of the number of instances per class.

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.

1: \( q = \text{sequence} \)
2: \( K = \{awc, aw, ac, wc, a, w, c\} \)
3: procedure IMPUTESEQUENCES\((q, K)\)
4: \( k = \text{Randomly select one element from } K \)
5: if \( k == awc \) then
6: \( T \leftarrow q \)
7: else if \( k == aw \) then
8: \( T \leftarrow \text{set chest values to zeros} \)
9: else if \( k == ac \) then
10: \( T \leftarrow \text{set wrist values to zeros} \)
11: else if \( k == wc \) then
12: \( T \leftarrow \text{set ankle values to zeros} \)
13: else if \( k == a \) then
14: \( T \leftarrow \text{set chest & wrist values to zeros} \)
15: else if \( k == w \) then
16: \( T \leftarrow \text{set chest & ankle values to zeros} \)
17: else if \( k == c \) then
18: \( T \leftarrow \text{set ankle & wrist values to zeros} \)
19: end if
20: return \( T \)
21: 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.
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.

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.](image)

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 scenario 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 misclassification, 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
Chapter 4. Experimental Setup and Results

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
Chapter 4. Experimental Setup and Results

29

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
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 whiskers 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 recurrent 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
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 feedforward 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.

\begin{verbatim}
1: X = GeneActiv measurements
2: E = energy expenditure values
3: t = number of time steps
4: r = sampling rate
5: procedure SEQUENCESGENERATOR(X, r, t, E)
6:   w ← empty list
7:   for each e in E do
8:     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 \( e \)
9:   end for
10: return w
11: end procedure
\end{verbatim}

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.

1: \( X = \text{GeneActiv} \) measurements  
2: \( E = \text{energy expenditure values} \)  
3: \( t = \text{number of time steps} \)  
4: \( r = \text{sampling rate} \)  
5: \( \textbf{procedure SEQUENCES GENERATOR}(X, r, t, E) \)  
6: \( w \leftarrow \text{empty list} \)  
7: \( \textbf{for each} \ e \in E \ \textbf{do} \)  
8: \( \text{Create sequence with } t \text{ time steps from } X \)  
9: \( \text{Split measurements in sequence into 4 equal segments} \)  
10: \( \text{Downsample the measurements in each segment using sampling rates in } r \)  
11: \( \textbf{end for} \)  
12: \( \textbf{return} \ w \)  
13: \( \textbf{end procedure} \)

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 \( w_i \), 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. \( \text{flat} \) - this model was trained with sequences transformed using algorithm 4 above.
2. \( \text{SMV_flat} \) - this model was trained with SMV sensor measurements transformed into sequences using algorithm 4 above.
3. \( \text{equalSegments} \) - this model is trained with sequences transformed using algorithm 5 above.
4. \( \text{SMV_equalSegments} \) - this model is trained with SMV sensor measurements transformed into sequences using algorithm 5 above.
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
Chapter 4. Experimental Setup and Results

<table>
<thead>
<tr>
<th></th>
<th>RMSE</th>
<th>R-squared</th>
</tr>
</thead>
<tbody>
<tr>
<td>flat</td>
<td>1.576</td>
<td>0.382</td>
</tr>
<tr>
<td>SMV_flat</td>
<td>1.487</td>
<td>0.474</td>
</tr>
<tr>
<td>equalSegment</td>
<td>1.518</td>
<td>0.430</td>
</tr>
<tr>
<td>SMV_equalSegment</td>
<td>1.464</td>
<td>0.467</td>
</tr>
</tbody>
</table>

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

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.
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 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, stackingShelves, 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

A.1 Confusion matrices

Figure A.1: Confusion matrix of the predictions from all models having wrist and chest sensor measurements.
**Figure A.2:** Confusion matrix of the predictions from all models having ankle and chest sensor measurements.
## Figure A.3: Confusion matrix of the predictions from all models having chest sensor measurements only.
**Figure A.4:** Confusion matrix of the predictions from all models having wrist sensor measurements.
Bibliography


[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.


