Detecting human activities based on a multimodal sensor data set using a bidirectional long short-term memory model: A case study

Silvano Ramos de Assis Neto, Guto Leoni Santos, Elisson da Silva Rocha, Malika Bendechache, Pierangelo Rosati, Theo Lynn, and Patricia Takako Endo

Abstract Human falls are one of the leading causes of fatal unintentional injuries worldwide. Falls result in a direct financial cost to health systems, and indirectly, to society's productivity. Unsurprisingly, human fall detection and prevention is a major focus of health research. In this chapter, we present and evaluate several bidirectional long short-term memory (Bi-LSTM) models using a data set provided by the Challenge UP competition. The main goal of this study is to detect 12 human daily activities (six daily human activities, five falls, and one post-fall activity) derived from multi-modal data sources - wearable sensors, ambient sensors, and vision devices. Our proposed Bi-LSTM model leverages data from accelerometer and gyroscope sensors located at the ankle, right pocket, belt, and neck of the subject. We utilize a grid search technique to evaluate variations of the Bi-LSTM model and identify a configuration that presents the best results. The best Bi-LSTM model achieved good results for precision and f1-score, 43.30% and 38.50%, respectively.

Silvano Ramos de Assis Neto
Universidade de Pernambuco, Pernambuco, Brazil. e-mail: silvano.neto@upe.br

Guto Leoni Santos
Universidade Federal de Pernambuco, Pernambuco, Brazil. e-mail: gls4@cin.ufpe.br

Elisson da Silva Rocha
Universidade de Pernambuco, Pernambuco, Brazil. e-mail: esr2@ecomp.poli.br

Malika Bendechache
Irish Institute of Digital Business, Dublin City University, Dublin, Ireland. e-mail: malika.bendechache@dcu.ie

Pierangelo Rosati
Irish Institute of Digital Business, Dublin City University, Dublin, Ireland e-mail: pierangelo.rosati@dcu.ie

Theo Lynn
Irish Institute of Digital Business, Dublin City University, Dublin, Ireland. e-mail: theo.lynn@dcu.ie

Patricia Takako Endo (Corresponding author)
Universidade de Pernambuco, Pernambuco, Brazil e-mail: patricia.endo@upe.br

1 Introduction

Falls are a major global public health problem. Research by the World Health Organisation (WHO) suggests that every year, approximately 37.3 million falls are severe enough to require medical attention and that falls are the second leading cause of fatal unintentional injuries (approx. 646,000 per annum), second only to road traffic injuries [1]. While older people have the highest risk of death or serious injury arising from a fall, children are also a high risk group for fall injury and death due to their stage of development associated characteristics and 'risk-taking' behaviors [1, 2]. Falls result in a significant direct financial cost to health systems, both in terms of in-patient and long term care costs, but also in indirect costs resulting from lost societal productivity of the focal person and caregivers [2]. To illustrate this impact, falls are estimated to be responsible for over 17 million lost disability-adjusted life years in productivity per annum [1]. Furthermore, fear of falling not only contributes to a higher risk of falling but can result in indirect negative health consequences including reduction or avoidance of physical activity and psychological issues, which can contribute to a lower quality of life [3].

Unsurprisingly, fall detection and prevention is a major focus of public health initiatives and research. Preventative initiatives include clinical interventions, environmental screening, fall risk assessment and modification, muscle strengthening and balance retraining, assistive devices, and education programs [1, 2]. Fall detection systems include non-wearable (sometimes referred to as context-aware systems) and wearable systems whose main objective is to alert when a fall event has occurred [4]. Research on fall detection systems suggests that these systems both reduce the fear of falling and actual falls as well as mitigating negative consequences of falls due to faster fall detection and intervention in the instance of a fall [5].

Advances in low-cost sensing devices and their integration into both mobile and so-called 'smart' environments have accelerated research into human activity recognition (HAR). Researchers are increasingly able to draw on a combination of wearable devices and fixed location data sources to inform HAR research efforts by providing different perspectives of a given event or human activity [6]. Making sense of this heterogeneous multi-modal data is not without challenges, not least those presented by the volume, variety, and velocity of such time-series data but also the specific human activity being explored and the efficacy of a given HAR technique [7, 8, 9, 10].

In this chapter, we present a deep learning model to detect falls using multi-modal sensor data. We propose a bidirectional long short-term memory (Bi-LSTM) model that leverages data accelerometer and gyroscope sensors located at the ankle, right pocket, belt, and neck of the subject. We propose two model configurations, one identified empirically and a second identified using a grid search technique.

The rest of this chapter is organized as follows. In Section 2, we describe the basic concepts of LSTM and Bi-LSTM. We then present the methodology applied in this study in Section 3, both describing the data set and the evaluation metrics. Section 4 describes our Bi-LSTM model and Section 5 presents the results achieved by our

models. Section 6 briefly presents related work. We conclude with a summary of our work and directions for further research in Section 7.

2 Long short-term memory (LSTM)

Deep learning networks, such as Multilayer Perceptron (MLP), Convolutional Neural Networks (CNN), and Radial Basis Function Networks amongst others, assume that all inputs are independent of each other. As such, they are not appropriate for time-series data related to human activities. Recurrent Neural Networks (RNNs) are able to overcome this limitation by using a recurrent connection in every neuron [11]. The activation of a neuron is fed back to the neuron itself in order to provide a memory of past activations and to learn the temporal dynamics of time-series data [11]. However, RNNs have limitations when it comes to discovering patterns over long temporal intervals [12] as they are subject to both exploding and vanishing gradient problems [13, 14]. While the former is relatively easy to address using gradient clipping [12, 15], vanishing gradient problems are more challenging [16]. Long short-term memory (LSTM) is a variant of traditional RNN which overcomes both problems [16]. LSTM networks make use of recurrent neurons with memory blocks, working with the concept of gates [17, 11]. While they overcome vanishing and exploding gradient problems, each unit of an LSTM requires intensive calculations resulting in long training times [14]. Figure 1 presents a basic schema of an LSTM block.

An LSTM network updates its block state according to gate activation. Thus, the input data provided to the LSTM network is fed into the gates that define which operation should be performed: write (input gate), read (output gate), or reset (forget gate). The mechanisms of these gates are based on component-wise multiplication of the input. The vectorial representation of each gate is as follows [11]:

$$i_t = \sigma_i(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \tag{1}$$

$$f_t = \sigma_f(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \tag{2}$$

$$c_t = f_t c_{t-1} + i_t \sigma_c(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{3}$$

$$o_t = \sigma_o(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \tag{4}$$

$$h_t = o_t \sigma_h(c_t) \tag{5}$$

where i, f, o, and c represent the outputs of input gate, forget gate, output gate, and cell activation vectors, respectively; all of them have the same vector size h_t therefore defining the hidden value (i.e., the memory state of the block). σ_i, σ_f, and

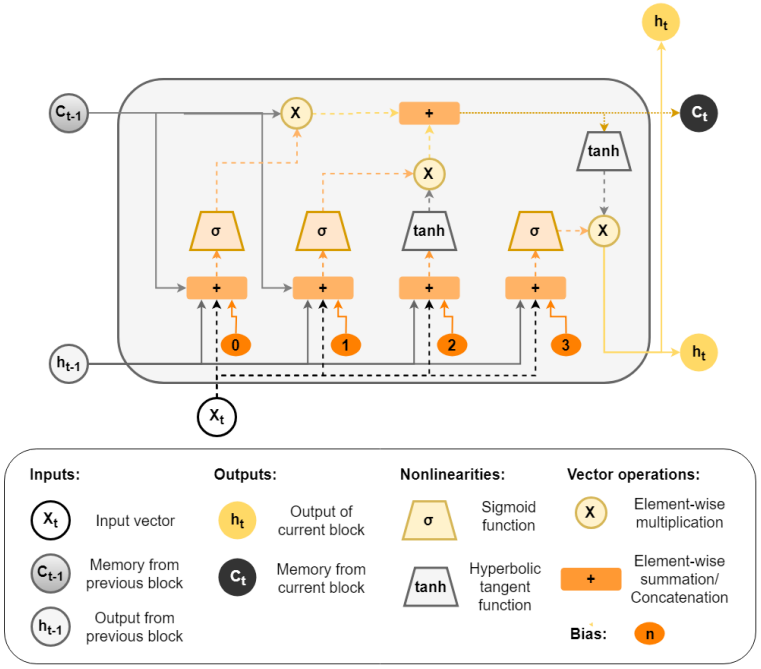

Fig. 1: Example of an LSTM block (adapted from [18])

σ_o are, respectively, the non-linear functions of input, forget, and output gates. W_{xi}, W_{hi}, W_{ci}, W_{xf}, W_{hf}, W_{cf}, W_{xc}, W_{hc}, W_{xo}, W_{ho}, and W_{co} are weight matrices of the respective gates, where x and h are the input and the hidden value of LSTM block respectively. b_i, b_f, b_c, and b_o are the bias vectors of input gate, forget gate, cell, and output gate, respectively [11].

The main difference between an RNN and a feed-forward model is the ability of the RNN to consider past information at each specific time step [19]. However, in some use cases, a wider context must be taken into account. In speech recognition, for example, the correct classification and interpretation of a given sound depends on the proceeding phoneme [20]. Correct classification of other data types, such as text and time-series, also depends on both preceding and subsequent data.

Bidirectional RNNs (Bi-RNNs) are able to process both past and future information at each time step [21]. In order to do so, each hidden layer of a Bi-RNN is composed of two hidden layers i.e. one for processing the past time steps and another for processing future time steps. The outputs are then combined to compose a new output that is forwarded to the next hidden layers [19]. Therefore, the output of each time step includes more complete clues related to the wider context of each specific input data. For the study described in this chapter, we use Bi-LSTM, a type of Bi-RNN.

3 Methodology

3.1 The data set

The data set used for this study, UP-Fall Detection, was made available as part of the Challenge UP: Multi-modal Fall Detection competition [22, 23]. The data set includes five falls and six daily activities performed by 12 subjects (see Table 1). Subjects performed five different types of human falls (falling forward using hands, falling forward using knees, falling backwards, falling from a sitting position on an empty chair and falling sideward), six simple human daily activities (walking, standing, picking up an object, sitting, jumping, and lying down), and an additional activity labeled as "on knees" where a subject remained on their knees after falling.

Table 1: Description of activities [22].

Activity ID	Description
1	Falling forward using hands
2	Falling forward using knees
3	Falling backwards
4	Falling sideward
5	Falling from sitting in a chair
6	Walking
7	Standing
8	Sitting
9	Picking up an object
10	Jumping
11	Lying Down
20	On knees

3.2 Data collection

The data was collected using a multi-modal approach from wearable sensors, ambient sensors, and vision devices distributed as per Figure 2. The experiments were conducted in a controlled laboratory environment in which light intensity did not vary; the ambient sensors and cameras remained in the same position during the data collection process.

For our study, we used data from five Mbientlab MetaSensor wearable sensors collecting raw data from a 3-axis accelerometer, a 3-axis gyroscope, and the ambient light value. These wearable sensors were located on the left wrist, under the neck, at the right trouser pocket, at the middle of the waist (on/in the belt), and at the left ankle. Also, data from one electroencephalograph (EEG) NeuroSky MindWave

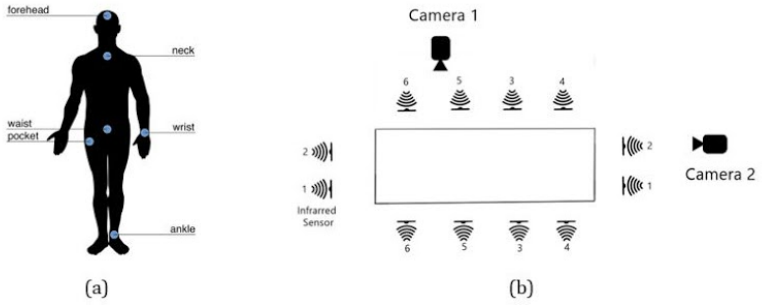

Fig. 2: Distribution of the sensors used to collect data: (a) Wearable sensors and EEG headset located on the human body, and (b) Layout of the context-aware sensors and camera views [22].

headset was used to measure the raw brainwave signal from a unique EEG channel sensor located at the forehead.

For context-aware sensors, six infrared sensors above the floor of the room measured the changes through interruption of the optical devices.

Lastly, two Microsoft LifeCam Cinema cameras were located above the floor, one for a lateral view and the other for a frontal view.

3.3 Evaluation metrics

For the Challenge UP competition [22], the F1-score measure was used to evaluate proposed models, considering both precision and sensitivity (recall). The F1-score is calculated as shown in Eq. 6:

$$F1 = 2 \times \frac{Precision_{\mu} \times Sensitivity_{\mu}}{Precision_{\mu} + Sensitivity_{\mu}} \tag{6}$$

where $Precision_{\mu}$ is the average number of the number of true positives (TP) across all activities and falls divided by the sum of true positives (TP) and false positives (FP) (Eq. 7); and $Sensitivity_{\mu}$ is the average number of TP across all activities and falls divided by the sum of TP and false negatives (FN) (Eq. 8).

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$Sensitivity = Recall = \frac{TP}{TP + FN} \tag{8}$$

In addition to the requirements of the Challenge UP competition outlined above, we also consider specificity and accuracy. While sensitivity is used to determine the proportion of actual positive cases predicted correctly and thus avoid false negatives, specificity is used to determine the proportion of actual negative cases predicted

correctly i.e. the avoidance of false positives. Together, sensitivity and specificity provide a more informed decision on the efficacy of a given model. Specificity is calculated as the average number of true negatives (TN) divided by the sum of TN and FP (Eq. 9).

$$Specificity = \frac{TN}{TN + FP} \tag{9}$$

Accuracy is a metric widely used to compare machine and deep learning models because it evaluate generally how many samples of test data were labeled correctly. Accuracy can be calculated as the average number of TP and TN across all activities and falls divided by the total number of cases examined (Eq. 10).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{10}$$

4 A Bi-LSTM model for human activities detection

We propose a Bi-LSTM model to identify human activities. The proposed empirical model (Figure 3) is composed of three bidirectional layers containing 200 LSTM cells each (above this, the model started to obtain worse results), interspersed by dropout layers with a 25% probability rate in order to reduce the overfitting of the model. Hyperbolic tangent and sigmoid were set as functions of activation and recurrent activation of these cells, respectively. The output layer is a fully connected layer with 12 units (the data set contains 12 different activities to be classified; see Table 1) with softmax activation function. Figure 4 presents the code that implements our Bi-LSTM model[1]. The model implementation was done using the Keras framework[2] with TensorFlow[3] as the backend.

As we are dealing with multi-label classification, we used categorical cross-entropy as a loss function [24]. Equation 11 illustrates the categorical cross-entropy function, where y is the array of real values, $\hat{y}$ is the array of predictions, N is the size of predictions, and M is the number of classes. The Adam algorithm as an optimizer [25].

$$L(y, \hat{y}) = -\sum_{j=0}^{M}\sum_{i=0}^{N}(y_{ij} * log(\hat{y}_{ij})) \tag{11}$$

The learning rate is equal to 0.001, β_1 and β_2 equal to 0.9 and 0.999, respectively. These parameters were defined empirically. For the training of this model, a pattern of at least 12 epochs was identified as the maximum reach of the network performance. Above 12 epochs, the performance tended to stabilize.

[1] The entire code is available for download at https://github.com/GutoL/ChallengeUP.

[2] http://keras.io/

[3] https://www.tensorflow.org/

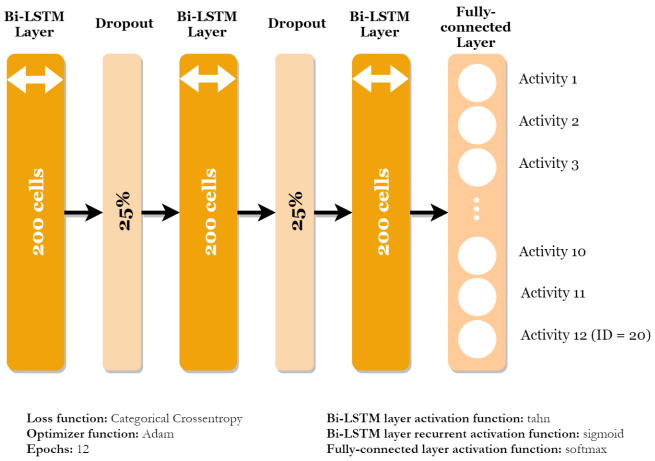

Fig. 3: Bidirectional LSTM Network

```
model = Sequential()


activation_Dense='softmax'
units = 200

#LSTM MODEL

model.add(Bidirectional(CuDNNLSTM(units=units,
                                  return_sequences=True,
                                  input_shape=input_shape)))

model.add(Dropout(rate=0.25))

model.add(Bidirectional(CuDNNLSTM(units=units,
                                  return_sequences=True)))

model.add(Dropout(rate=0.25))

model.add(Bidirectional(CuDNNLSTM(units=units,
                                  return_sequences=True)))

model.add(Flatten())

model.add(Dense(12,activation=activation_Dense))


adam = optimizers.Adam(lr=0.001,
                       beta_1=0.9,
                       beta_2=0.999,
                       epsilon=None,
                       decay=0.0,
                       amsgrad=False)

model.compile(loss = 'categorical_crossentropy',
              optimizer = adam)
```

Fig. 4: Code that implements our Bi-LSTM model

4.1 Data pre-processing

In order to fit the data set to our model, we performed data pre-processing. Firstly, we sampled the data set on a per second basis (totaling 9,187 samples). The number of data points per second was not constant in our data set as each sensor may have collected data at different points in time. Thus, we used data padding in order to generate samples with a similar size. Thence, we considered the length of the greater sample (the second with more data points i.e. 22 points) and applied the padding; this was repeated until the sample comprised 22 points. In the end, the data set comprised 9,187 complete samples.

Finally, we divided the data set in to two parts, allocating 80% (7,349 samples) for training and 20% (1,838 samples) for testing, an approach widely used in the literature [26] [27].

5 Results

5.1 Selecting the best Bi-LSTM model

We utilized a grid search technique to evaluate different Bi-LSTM architectures and then selected the best performing architecture for further evaluation.

Grid search identifies tuples from the combination of suggested values for two or more parameters, trains the model for each possible combination and compares the results of a predefined metric. Despite some limitations (see [28] for a more detailed discussion), grid search still represents the state of the art for hyper-parameter optimization and has been adopted in several machine learning studies (e.g., [29], [30], [31] and [32]).

As shown in Table 2, we defined different levels for different parameters of the model. The grid search was run 10 times and the average of all metrics was calculated in order to take into account the variation of results due to the stochastic nature of the optimization process [25].

Table 2: Parameters and levels

Parameters	**Levels**
Number of layers	From 1 to 3, step 1
Number of nodes	From 100 to 250, step 25

Figures 5, 6, 7, 8, and 9 show the results for accuracy, precision, recall, specificity, and f1-score for all model configurations used in grid search, respectively.

Regarding **accuracy** (Figures 5), the best model configuration uses 1 layer and 250 units, reaching, on average, 70%; while the model configuration with 3 layers and 200 units obtained the worst result, 65.9%, on average.

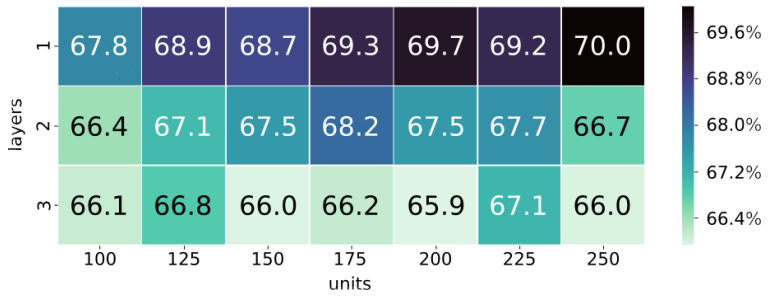

Fig. 5: Accuracy results for all model configurations used in the grid search approach.

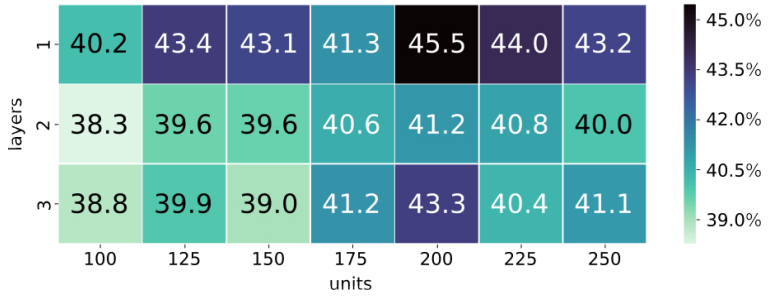

Fig. 6: Precision results for all model configurations used in the grid search approach.

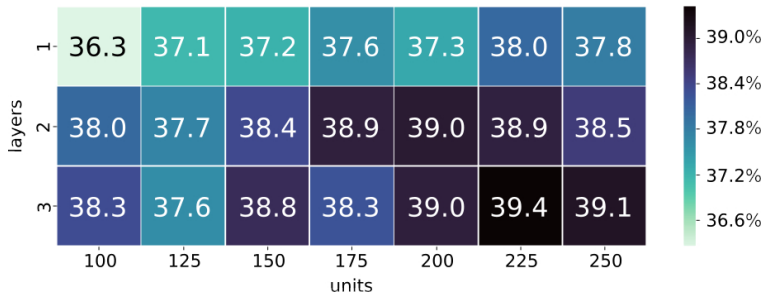

Fig. 7: Sensitivity results for all model configurations used in the grid search approach.

For **precision** (Figure 6), the model configuration that gives, on average, the best result was 1 layer and 200 units (45.5%); while the worst average precision result (38.3%) was obtained by the model configuration with 2 layers and 100 units.

The **sensitivity** results (Figure 7) suggest the configuration with 3 layers and 225 units presented the best recall result, achieving, on average, 39.4%; and the simplest model configuration, with 1 layer and 100 units presented the worst result, 36.3%, on average.

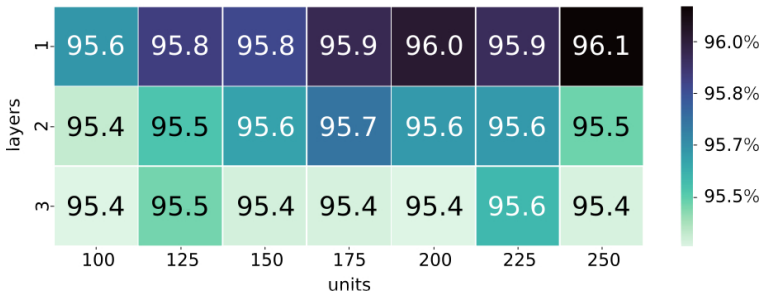

Fig. 8: Specificity results for all model configurations used in the grid search approach.

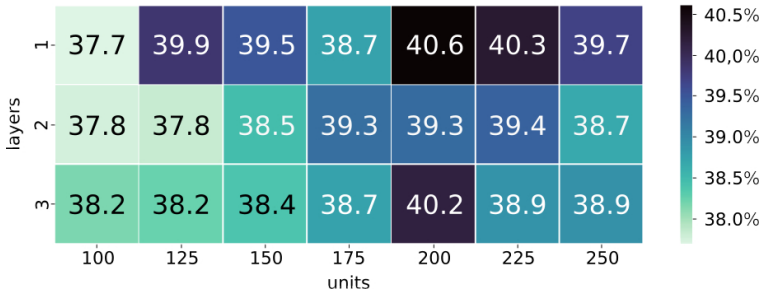

Fig. 9: F1-score results for all model configurations used in the grid search approach.

The **specificity** results, as shown in Figure 8, suggest the model configuration that obtained the best result was composed of 1 layer and 250 units, achieving 96.10% specificity, on average. On the other hand, for this metric, different model configurations obtained the same worst level of specificity, on average, 95.40%. Such models were: 3 layers and 100, 150, 175, 200, and 250 units; and 2 layers and 100 units obtained.

Finally, considering the **f1-score**, as illustrated in Figure 9, the model configuration that presented the best result was 1 layer and 200 units, with 40.60%, on average. The model with 1 layer and 225 units also achieved a good f1-score, 40.30%, and the model with 3 layers and 200 units found 40.20%. On the other hand, the model configuration that obtained the worst f1-score level was the simplest configuration, with 1 layer and 100 units, achieving 37.70%, on average; followed by the models with 2 layers and 100 and 125 units, both with 37.80%.

From the grid search results, one can note that there is no common behavior when analyzing the best performance per metric, meaning that for each metric, a different model can achieve the best result. The only exception was the model with 1 layer and 250 units, that found the best results for accuracy and specificity metrics.

The best model configuration in terms of accuracy (Figure 5) uses 1 layer and 250 units, reaching, on average, 70%; while the model configuration with 3 layers and 200 units obtained the worst result with an average accuracy score of 65.9%. For

precision (Figure 6), the model configuration that gives the best performing model uses 1 layer and 200 units (average score 45.5%) while the worst model (average score 38.3%) uses 2 layers and 100 units. For sensitivity (Figure 7), there seems to be a positive relationship between complexity and average sensitivity score with the simplest model configuration (1 layer and 100 units) showing the worst performance (average score 36.3%) with the second most complex model (3 layers and 225 units) providing the best results (average score 39.4%). In the case of specificity (Figure 8), the model configuration that provides the best result uses 1 layer and 250 units (average score 96.10%) while a number of different configurations demonstrated poor performance (average score 95.40%). Finally, for the f1-score (Figure 9), the model configuration using 1 layer and 200 units, the model using 1 layer and 225 units, and the model using 3 layers and 200 units, achieved similar results (average score 40.60%, 40.30%, and 40.20% respectively). On the other hand, the simplest model configuration (1 layer and 100 units) achieving the worst results (average score of 37.70%).

Interestingly, the results of the grid search suggest that there is no single model specification with consistently superior performance across different metrics. The model configurations achieving the best results according to one metric did not provide comparable results for any other metric. The only exception is the model with 1 layer and 250 units, that provides the best results for accuracy and specificity.

Another interesting observation relates to the number of layers in the model. In deep learning models, the concept of "depth" is related to the presence of several interconnected hidden layers with tens or hundreds of neurons. However, based on the results of our experiment, adding additional hidden layers does not always result in better model performance. For instance, the best result in terms of precision was obtained with 1 layer and 200 units (Figure 6). Increasing the number of layers to 3, resulted in a 2.2% decrease in precision. A similar relationship appears across all other metrics with the only exception being sensitivity, where models with 3 layers tended to provide better results.

It is also worth highlighting that specificity is the metric with the highest average values, while sensitivity has the lowest. This suggests that our models are more able to predict true positives than true negatives.

Due to time constraints, we did not perform the grid search when initially designing the model submitted to the Challenge UP competition. Consequently, we present the results from two model configurations below, one identified empirically (Challenge UP results) and a second identified using grid search.

5.2 Challenge UP results

Figure 10 presents the confusion matrix regarding the test results using the Bi-LSTM model presented in Challenge UP. The model did not obtain good results in predicting falls (activities from 1 to 5). For example, all samples of activities 1 (falling forward using hands) and 3 (falling backwards) were misclassified by the

model. This occurred because the data set has few samples of falls, and deep learning models perform better with data sets that contain larger sample sizes.

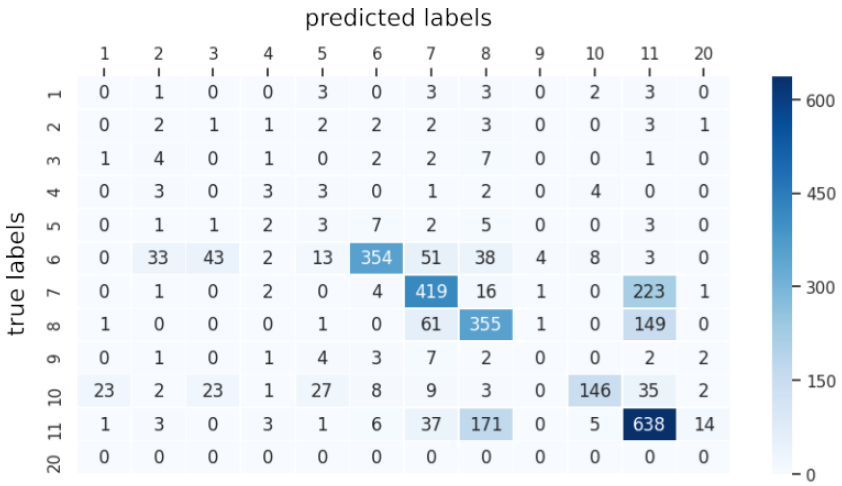

Fig. 10: Confusion matrix of the Challenge UP model

The Bi-LSTM model achieved the best results with classes that have more samples. Class 11 (lying down) achieved 638 correct predictions, followed by Class 7 (standing) with 419 hits. These classes obtained better results due to the simplicity of the activities captured by the sensors - the person remains still i.e. without making any movement. However, for those same classes (11 and 7), the model also misclassified a lot. For Class 11, the model misclassified 171 samples as Class 8 (sitting); and for Class 7, the model misclassified 223 samples as Class 11 (lying down).

Table 3 shows the evaluation results (in percentages) for precision, sensitivity, specificity, and f1-score for each activity presented in Table 1. Note that accuracy is not measured in this case because it is a global metric.

The model obtained good specificity results for all classes, achieving the best results for Class 9 (99.69%). Similarly, the model has a f1-score of 0% for Classes 1, 3, 9 and 20. However, for the other metrics, poor results were achieved. For example, for Classes 1, 3, 9, and 20, the value of the precision, sensitivity, and f1-score was 0%. In contrast, Classes 6, 7, 8, 10, and 11 achieved the best results for f1-score - 75.72%, 66.46%, 60.53%, 65.77%, and 65.81%, respectively. This is explained by the greater volume of samples for these classes in the data set.

One can see from Table 3 that the most critical metric for the Bi-LSTM model is sensitivity, which corresponds to the true positive rate. As the model misclassified several samples (Figure 10), the overall sensitivity results are considered poor. Class 11 returned the highest sensitivity rate because it was the most correctly classified class in the data set. Table 4 presents the overall results for accuracy, precision, sensitivity, and f1-score for the model presented in Challenge UP. One can see that

Table 3: Evaluation results (in %) for all activities using the Bi-LSTM model presented in Challenge UP competition

Activity	Precision	Sensitivity	Specificity	F1-score
1	0	0	98.6639	0
2	3.9216	11.7647	97.5089	5.8824
3	0	0	96.5795	0
4	18.7500	18.7500	99.3264	18.7500
5	5.2632	12.5000	97.2603	7.4074
6	91.7098	64.4809	97.9975	75.7219
7	70.5387	62.8186	89.5585	66.4552
8	58.6777	62.5000	86.2259	60.5286
9	0	0	99.6885	0
10	88.4848	52.3297	98.9403	65.7658
11	60.1887	72.5825	75.2347	65.8071
20	0	0	98.9691	0

the model achieved 62.89% for accuracy, while other metrics achieved c. 32%. Since the data set used was unbalanced, the model classified the classes with more samples correctly (see Table 3).

Table 4: Overall metrics of the Bi-LSTM model presented in Challenge UP competition

Metrics	Results
Overall accuracy	62.89%
Mean global precision	33.13%
Mean global sensitivity	32.52%
Global f1-score	32.82%

5.3 Bi-LSTM model results (grid search)

Figure 11 presents the confusion matrix for the best model configuration found by the grid search based on F1-score i.e. 1 layer and 200 units. The model did not obtain good results in predicting falls (activities from 1 to 5). In fact, activities 1 to 5 were largely misclassified. This is most likely related to the limited number of falls in the data set; deep learning models perform better with large samples. The Bi-LSTM model achieved the best results with classes that had more samples. For example, for Class 11 (lying down) and Class 7 (standing), the model generated 693 and 450 correct predictions respectively.

Table 5 presents the evaluation results (in %) for precision, sensitivity, specificity, and f1-score for each activity. Note that accuracy is not measured in this case because it is a global metric.

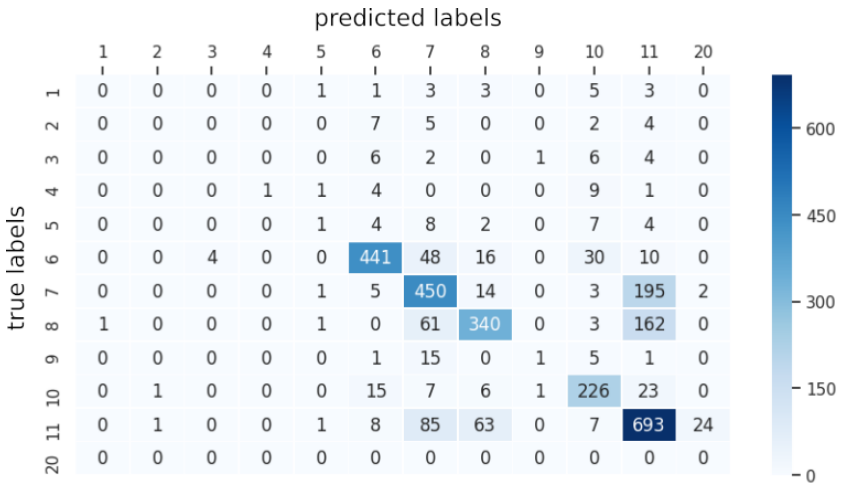

Fig. 11: Confusion matrix of the best model configuration found by the grid search

In, general, the model achieved very good specificity results for all classes (achieving 100% when considering Class 4). However, the model performed poorly for Classes 1, 2, 3, and 20 presenting a score of 0% for precision, sensitivity, and f1-score.

Table 5: Evaluation results (in %) for all activities using the best Bi-LSTM model found by the grid search

Activity	**Precision**	**Sensitivity**	**Specificity**	**F1-score**
1	0	0	99.9536	0
2	0	0	99.9072	0
3	0	0	99.8146	0
4	100	6.2500	100	11.7647
5	16.6667	3.8462	99.7682	6.2500
6	89.6341	80.3279	97.1072	84.7262
7	65.7895	67.1642	87.9195	66.4697
8	76.5766	59.8592	94.5749	67.1937
9	33.3333	4.3478	99.9071	7.6923
10	74.5875	81.0036	96.1577	77.6632
11	63	78.5714	78.2003	69.9294
20	0	0	98.8068	0

These results illustrate some weaknesses of the proposed Bi-LSTM model configurations when working with an unbalanced data set. The classes that presented metrics equal to 0% were classes comprising relatively small samples.

Finally, Table 6 presents the overall results for accuracy, precision, sensitivity, and f1-score of the proposed model configuration. Similar to the results of the Challenge

UP model, the model obtained the best result for accuracy (70.22%) when compared to the other metrics, reflecting the previously discussed uneven samples in the data set. The overall precision score was 43.30%; the sensitivity score was 34.67%, and the f1-score was 38.50%. One can note an improvement in all metrics from the initial model presented in Challenge UP to the revised model obtained using the grid search method.

Table 6: Overall metrics for the best Bi-LSTM model configuration found by the grid search

Metrics	**Results**
Overall accuracy	70.22%
Mean overall precision	43.30%
Mean overall sensitivity	34.67%
Overall f1-score	38.50%

6 Related work

Human activity recognition (HAR) can play an important role in people's daily lives due to its ability to learn important high-level knowledge about human activity from raw sensor inputs [33]. The increasing popularity of HAR is correlated with the diversity and popularity of wearable and on-body sensing devices such as accelerometers, gyroscopes, sound sensors, and image capture devices amongst others. HAR has drawn extensive attention in health and computer science research and is playing an increasingly important role in various research areas including home behaviour analysis [34], health monitoring [35], and gesture recognition [36].

There is a well established literature on HAR using machine learning. Historically, many studies focused on data with a single modality such as single sensor-based data [37, 38, 39, 33]. Single modality data is inherently limited for HAR studies in real-world settings due to high intra-class and low inter-class variations in the actions performed for a particular application [10]. Therefore, to exploit the benefits of machine learning techniques for a learning-based HAR, it is extremely important to have multi-modal data sets [33]. Multi-modal machine learning aims to build models that can process and relate information from multiple modalities [40].

More recently, there has been an increasing focus on the study of learning-based HAR using multi-modal data, and in particular, multi-modal time-series data. Existing methods can be divided into two categories: shallow learning-based HAR and deep learning-based HAR. The former relies on extracting a set of features from time-series sensor signals and mapping these handcrafted features to various human activities. Subsequently, a shallow supervised machine learning algorithm is applied to recognize activities. The most popular learning algorithms include decision trees [41, 42], K Nearest Neighbour (KNN) [43, 44], and Support Vector Machines (SVM) [45, 46]. For example, [46] extracts 561 features from an accelerometer and

gyroscope, and applies a multi-class SVM to classify six different activities. The common characteristic of these methods is that they perform feature extraction manually which is task-dependent and requires human intervention, thereby impacting effectiveness. As a result, many researchers have turned their attention to deep learning approaches for automatic feature extraction. At the same time, implicit features can be learned by models that may not be possible using manual or handcrafted methods [47].

Many different deep learning models have been used to recognize human activities in a wide range of contexts including CNN, RNN, and particularly in the context of this chapter, LSTM networks. A very recent paper [48] proposed a baseball player behavior classification system using LSTM that accurately recognizes many baseball player behaviors. The classifier is trained on multi-modal data collected from multiple heterogeneous IoT sensors and cameras. [49] also used an LSTM network to detect daily human activities including eating and driving activity. The authors adopted a two-level ensemble model to combine class-probabilities of multiple sensor modalities, and demonstrated that a classifier-level sensor fusion technique for multi-modality can improve the classification performance compared to single modality data.

Authors in [50] used LSTM in a biometrics application to identify individual humans based on their motion patterns captured from smartphone features i.e. accelerometer, gyroscope and magnetometer data. The use of LSTM demonstrated that human movements convey necessary information about the person's identity and it is possible to achieve relatively good authentication results. The authors also demonstrated that the same LSTM algorithm can also be applied to other time-series data e.g. for gesture detection in a human conversation. In [51], inertial signals from a set of wearable sensors were used and fed as images into a CNN network to recognize human activities. Using both CNN and LSTM as a hybrid model, authors in [11] classified human activities. They used CNN to automatically extract spatial features from raw sensor signals, and LSTM to capture the temporal dynamics of the human movement.

Several surveys on recent advances on deep learning methods for multi-modal HAR have been completed and are worth reviewing for those interested in the domain [7, 8, 9, 10].

While significant progress has been made, HAR remains a challenging task. This is partly due to the broad range of human activities as well as the rich variation in how a given activity can be performed. Deep learning shows great potential for a high-level abstraction of data. Therefore, more deep learning models need to be developed as self-configurable frameworks for HAR [47]. In this chapter, we propose a Bi-LSTM deep learning model to detect twelve types of human daily activities, and in particular, human falls. We use a multi-modal sensors data set generated from three different sources(i.e. wearable sensors, ambient sensors, and vision devices).

7 Conclusions

In this chapter, we propose a Bi-LSTM model to detect five different types of falls, six common daily human activities, and one post-fall activity. The data set was provided by the Challenge UP competition and was collected using a multi-modal approach generated from wearable sensors, ambient sensors, and vision devices.

Our Bi-LSTM model makes use of two wearable sensors (accelerometer and gyroscope) located at the ankle, right pocket, belt, and neck of the subject. In the training phase, the model was able to make good predictions of a selection of specific human activities (walking, standing, sitting, jumping and lying down). Our model is able to identify when a subject is lying down (when a fall has occurred) but it does not detect the type of fall (forward using hands, forward using knees, backwards, sideward, or falling after sitting on a chair). This result can be explained by the uneven samples of data by activity in the data set.

Future studies may explore other deep learning models, such as bidirectional gated recurrent units (bi-GRU), a simplified version of LSTM layers, or CNNs, and compare the sensitivity, specificity, precision, and accuracy of a range of different models. Given the limitations of the data set used in this study and the impact on results, larger data sets with sufficiently large samples of each activity are required for wider use. Future research may involve creating actual or synthetic data sets to address these needs and leverage these and other multi-modal datasets (e.g. cameras and environmental sensors) for further study.

Acknowledgements This work is partly funded by the Irish Institute of Digital Business (dotLAB).

References

[1] World Health Organization. Falls, Fact Sheet. `http://www.who.int/news-room/fact-sheets/detail/falls` (2018). Accessed: 2018-10-08

[2] W.H. Organization, *WHO Global Report on Falls: Prevention in Older Age* (World Health Organization, 2007)

[3] A.C. Scheffer, M.J. Schuurmans, N. Van Dijk, T. Van Der Hooft, S.E. De Rooij, Fear of falling: measurement strategy, prevalence, risk factors and consequences among older persons, Age and ageing **37**(1), 19 (2008)

[4] R. Igual, C. Medrano, I. Plaza, Challenges, issues and trends in fall detection systems, Biomedical engineering online **12**(1), 66 (2013)

[5] S. Brownsell, M.S. Hawley, Automatic fall detectors and the fear of falling, Journal of telemedicine and telecare **10**(5), 262 (2004)

[6] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, A.Y. Ng, Multimodal deep learning, in *Proceedings of the 28th international conference on machine learning (ICML-11)* (2011), pp. 689–696

[7] P.V. Rouast, M. Adam, R. Chiong, Deep learning for human affect recognition: Insights and new developments, IEEE Transactions on Affective Computing (2019)

[8] T. Baltrušaitis, C. Ahuja, L.P. Morency, Multimodal machine learning: A survey and taxonomy, IEEE Transactions on Pattern Analysis and Machine Intelligence **41**(2), 423 (2019)

[9] H.F. Nweke, Y.W. Teh, G. Mujtaba, M.A. Al-Garadi, Data fusion and multiple classifier systems for human activity detection and health monitoring: Review and open research directions, Information Fusion **46**, 147 (2019)

[10] C. Chen, R. Jafari, N. Kehtarnavaz, A survey of depth and inertial sensor fusion for human action recognition, Multimedia Tools and Applications **76**(3), 4405 (2017)

[11] F. Ordóñez, D. Roggen, Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition, Sensors **16**(1), 115 (2016)

[12] Y. Bengio, P. Simard, P. Frasconi, et al., Learning long-term dependencies with gradient descent is difficult, IEEE transactions on neural networks **5**(2), 157 (1994)

[13] S. Hochreiter, The vanishing gradient problem during learning recurrent neural nets and problem solutions, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems **6**(02), 107 (1998)

[14] H.Y. Lin, Y.L. Hsueh, W.N. Lie, Abnormal event detection using microsoft kinect in a smart home, in *Computer Symposium (ICS), 2016 International* (IEEE, 2016), pp. 285–289

[15] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation **9**(8), 1735 (1997)

[16] R. Jozefowicz, W. Zaremba, I. Sutskever, An empirical exploration of recurrent network architectures, in *International Conference on Machine Learning* (2015), pp. 2342–2350

[17] K. Greff, R.K. Srivastava, J. Koutník, B.R. Steunebrink, J. Schmidhuber, Lstm: A search space odyssey, IEEE transactions on neural networks and learning systems **28**(10), 2222 (2017)
[18] Understanding lstm and its diagrams. https://medium.com/mlreview/understanding-lstm-and-its-diagrams-37e2f46f1714 (2016). Accessed: August, 2018
[19] R. Zhao, R. Yan, J. Wang, K. Mao, Learning to monitor machine health with convolutional bi-directional lstm networks, Sensors **17**(2), 273 (2017)
[20] I. Goodfellow, Y. Bengio, A. Courville, Y. Bengio, *Deep learning*, vol. 1 (MIT press Cambridge, 2016)
[21] M. Schuster, K.K. Paliwal, Bidirectional recurrent neural networks, IEEE Transactions on Signal Processing **45**(11), 2673 (1997)
[22] Challenge up website. Available at: https://sites.google.com/up.edu.mx/challenge-up-2019/overview?authuser=0.Lastaccess: April,2019. (2019)
[23] L. Martínez-Villaseñor, H. Ponce, J. Brieva, E. Moya-Albor, J. Núñez-Martínez, C. Peñafort-Asturiano, Up-fall detection dataset: a multimodal approach, Sensors **19**(9), 1988 (2019)
[24] K. Zhao, W.S. Chu, H. Zhang, Deep region and multi-label learning for facial action unit detection, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 3391–3399
[25] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014)
[26] R.v.d. Berg, T.N. Kipf, M. Welling, Graph convolutional matrix completion, arXiv preprint arXiv:1706.02263 (2017)
[27] A. Mathis, P. Mamidanna, K.M. Cury, T. Abe, V.N. Murthy, M.W. Mathis, M. Bethge, Deeplabcut: markerless pose estimation of user-defined body parts with deep learning. Tech. rep., Nature Publishing Group (2018)
[28] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization, Journal of Machine Learning Research **13**(Feb), 281 (2012)
[29] J. Li, C. Zhang, Z. Li, Battlefield target identification based on improved grid-search svm classifier, in *2009 International Conference on Computational Intelligence and Software Engineering* (IEEE, 2009), pp. 1–4
[30] J.Y. Hesterman, L. Caucci, M.A. Kupinski, H.H. Barrett, L.R. Furenlid, Maximum-likelihood estimation with a contracting-grid search algorithm, IEEE transactions on nuclear science **57**(3), 1077 (2010)
[31] B. Zoph, Q.V. Le, Neural architecture search with reinforcement learning, arXiv preprint arXiv:1611.01578 (2016)
[32] M.F. Akay, Support vector machines combined with feature selection for breast cancer diagnosis, Expert systems with applications **36**(2), 3240 (2009)
[33] J. Wang, Y. Chen, S. Hao, X. Peng, L. Hu, Deep learning for sensor-based activity recognition: A survey, Pattern Recognition Letters **119**, 3 (2019)
[34] P. Vepakomma, D. De, S.K. Das, S. Bhansali, A-wristocracy: Deep learning on wrist-worn sensing for recognition of user complex activities, in *2015 IEEE*

12th International Conference on Wearable and Implantable Body Sensor Networks (BSN) (IEEE, 2015), pp. 1–6

[35] L. Ding, W. Fang, H. Luo, P.E. Love, B. Zhong, X. Ouyang, A deep hybrid learning model to detect unsafe behavior: integrating convolution neural networks and long short-term memory, Automation in Construction **86**, 118 (2018)

[36] J.C. Núñez, R. Cabido, J.J. Pantrigo, A.S. Montemayor, J.F. Vélez, Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition, Pattern Recognition **76**, 80 (2018)

[37] K. Altun, B. Barshan, Human activity recognition using inertial/magnetic sensor units, in *International workshop on human behavior understanding* (Springer, 2010), pp. 38–51

[38] M. Ermes, J. Pärkkä, J. Mäntyjärvi, I. Korhonen, Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions, IEEE transactions on information technology in biomedicine **12**(1), 20 (2008)

[39] G. Lefebvre, S. Berlemont, F. Mamalet, C. Garcia, Blstm-rnn based 3d gesture classification, in *International conference on artificial neural networks* (Springer, 2013), pp. 381–388

[40] A. Jaimes, N. Sebe, Multimodal human–computer interaction: A survey, Computer vision and image understanding **108**(1-2), 116 (2007)

[41] T. Van Kasteren, A. Noulas, G. Englebienne, B. Kröse, Accurate activity recognition in a home setting, in *Proceedings of the 10th international conference on Ubiquitous computing* (ACM, 2008), pp. 1–9

[42] J.R. Kwapisz, G.M. Weiss, S.A. Moore, Activity recognition using cell phone accelerometers, ACM SigKDD Explorations Newsletter **12**(2), 74 (2011)

[43] S. Hasan, M. Masnad, H. Mahmud, M. Hasan, Human activity recognition using smartphone sensors with context filtering, in *Proc. Ninth International Conference of Advances in Computer-Human Interactions* (2016), pp. 67–73

[44] K. Kunze, P. Lukowicz, Dealing with sensor displacement in motion-based onbody activity recognition systems, in *Proceedings of the 10th international conference on Ubiquitous computing* (ACM, 2008), pp. 20–29

[45] A. Bulling, D. Roggen, Recognition of visual memory recall processes using eye movement analysis, in *Proceedings of the 13th international conference on Ubiquitous computing* (ACM, 2011), pp. 455–464

[46] D. Anguita, A. Ghio, L. Oneto, X. Parra, J.L. Reyes-Ortiz, A public domain dataset for human activity recognition using smartphones., in *Esann* (2013)

[47] E. Kanjo, E.M. Younis, C.S. Ang, Deep learning analysis of mobile physiological, environmental and location sensor data for emotion detection, Information Fusion **49**, 46 (2019)

[48] S.W. Sun, T.C. Mou, C.C. Fang, P.C. Chang, K.L. Hua, H.C. Shih, Baseball player behavior classification system using long short-term memory with multimodal features, Sensors **19**(6), 1425 (2019)

[49] S. Chung, J. Lim, K.J. Noh, G. Kim, H. Jeong, Sensor data acquisition and multimodal sensor fusion for human activity recognition using deep learning, Sensors **19**(7), 1716 (2019)

[50] N. Neverova, C. Wolf, G. Lacey, L. Fridman, D. Chandra, B. Barbello, G. Taylor, Learning human identity from motion patterns, IEEE Access **4**, 1810 (2016)
[51] J. Yang, M.N. Nguyen, P.P. San, X.L. Li, S. Krishnaswamy, Deep convolutional neural networks on multichannel time series for human activity recognition, in *Twenty-Fourth International Joint Conference on Artificial Intelligence* (2015)