MIPRO 2018, May 21-25, 2018, Opatija Croatia

Real Time Human Activity Recognition on Smartphones using LSTM Networks

Martin Milenkoski, Kire Trivodaliev, Slobodan Kalajdziski, Mile Jovanov, Biljana Risteska Stojkoska
Faculty of Computer Science and Engineering (FCSE)
University “Ss. Cyril and Methodius”
Skopje, Macedonia
martin.milenkoski@students.finki.ukim.mk, kire.trivodaliev@ finki.ukim.mk, slobodan.kalajdziski@finki.ukim.mk,
mile.jovanov@finki.ukim.mk, biljana.stojkoska@finki.ukim.mk

Abstract— **Activity detection is becoming an integral part of many mobile applications. Therefore, the algorithms for this purpose should be lightweight to operate on mobile or other wearable device, but accurate at the same time. In this paper, we develop a new lightweight algorithm for activity detection based on Long Short Term Memory networks, which is able to learn features from raw accelerometer data, completely bypassing the process of generating hand-crafted features. We evaluate our algorithm on data collected in controlled setting, as well as on data collected under field conditions, and we show that our algorithm is robust and performs almost equally good for both scenarios, while outperforming other approaches from the literature.**

Keywords— activity recognition; LSTM, smartphone; wearable

I. Introduction

Physical activity duration, intensity and frequency are major lifestyle factors associated with beneficial health effects across the life span [1]. Therefore, activity detection is becoming the most important part of many healthcare applications, ranging from epidemiological and clinical studies to smartphone based “stay-fit” and “weight loss” applications.

Traditional way to measure activity is by attaching special hardware devices on predefined location, like hip and ankle. Sensor measurements from those devices are recorded on internal memory, to be later analyzed for different purposes [2]. Many of the epidemiological and clinical studies still use this method in their research [1]. With the technology improvement, body sensor networks allow more advanced approach, where sensor measurements can be sent directly to the users’ smartphone to be analyzed on the fly [3]. In the last few years, modern smartphones are equipped with dozens of different sensors, therefore, smartphone measurements can be used for the process of activity detection, bypassing the need for extra hardware devices [4].

There are many research in the literature that intend to perform activity detection and recognition from smartphone data [5]-[8]. In [5], different classification methods are applied (Multilayer Perceptron, Random Forest, etc.), achieving an overall accuracy rate of 91.15%. In [6], autoregressive coefficients, signal magnitude area and Kernel Discriminant Analysis are used to extract the features, while artificial neural nets are used for classification, achieving average accuracy of about 96%. Hardware-friendly approach in [7] adapts the standard Support Vector Machine (SVM) to reduce computational cost while maintaining accuracy comparable to other traditional SVM based classification methods. More recent approaches are focused on features extraction from the raw acceleration data. In [8], an unsupervised classification method is used for activity recognition.

The challenge in designing such algorithm is not only the accuracy of the algorithm, but also its computation cost, since it should operate on smartphone in real time [9]. Although modern smartphones have performances comparable with those of the computers, the power remains a challenging problem, since battery technology has not kept pace with information and communication technologies. Other issues regarding energy consumption is sampling frequency, as it is an important parameter for the accuracy of the algorithm.

The man goal of this paper was to develop a new lightweight algorithm for activity recognition, with the following characteristics: *(i)* to be easily implementable on mobile applications; *(ii)* to outperform other approaches from the literature by means of accuracy; and *(iii)* to be robust enough to perform almost equally good on data collected under field conditions as on data collected in a controlled environment. Our algorithm is based on neural network, i.e. Long Short Term Memory networks, which is able to learn features from raw accelerometer data, completely bypassing the process of generating hand-crafted features. Although this algorithm has been previously used for activity recognition [10], to the best of our knowledge, this is the first research that evaluates the algorithm on data collected from smartphone sensors.

The rest of this paper is organized as follows. In the next section, we explain the algorithm used for activity recognition, as well as tools used for its implementation as part of a mobile application. In the third section, we elaborate the data used for evaluation of our algorithm. Section four discusses the results. This paper is concluded in section five.

II. System Implementation

Although there are many techniques in the literature for activity recognition, in this paper we investigated a neural network approach based on Long Short Term Memory (LSTM) networks.

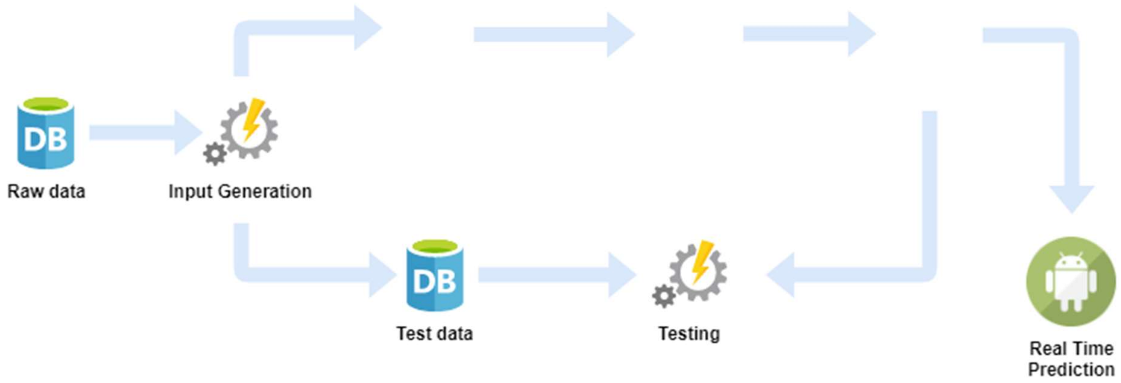


Fig. 1 Computation pipeline

LSTM network as a deep learning system is appropriate for temporal modeling and has shown improvements over Deep Neural Networks for speech recognition problem [11]. It was initially proposed by Hochreiter [12] and later improved in 2000 by Gers [13]. Since 2016, LSTM became integral part of many applications and services delivered by Google, Microsoft and Apple, including personalized speech recognition on smartphones [14] and gesture typing decoding [15]. In [10], LSTM is used for offline activity recognition, using different sensors from wearables. The authors identify that LSTM is suitable for multimodal wearables and does not require expert knowledge in designing features. Still, they do not implement LSTM on wearable devices.

In this section, a brief mathematical introduction of LSTM is given, followed by the computational procedure and software tools used for its implementation. Additionally, an Android mobile application was developed, explained in the last subsection.

A. Mathematical background

Long Short Term Memory (LSTM) networks are a special type of neural networks that remember information from further back in the past. Given a sequence of inputs X = $\{x_1, x_2, ..., x_n\}$, LSTM associates each time step with an input gate, memory gate and output gate, denoted respectively as i_t, f_t and o_t. The information from the past is remembered using the state vector c_{t-1}. The forget gate decides how much of the previous information is going to be forgotten. The input gate decides how to update the state vector using the information from the current input. The l_t vector consists of the information from the current input added to the state. Finally, the output gate decides what information to output at the current time step. This process is formalized as in (1),

$$\begin{aligned} i_t &= \sigma(W_i \cdot [h_{t-1}, x_t]) \\ f_t &= \sigma(W_f \cdot [h_{t-1}, x_t]) \\ o_t &= \sigma(W_o \cdot [h_{t-1}, x_t]) \\ l_t &= \tanh(W_l \cdot [h_{t-1}, x_t]) \\ c_t &= f_t \cdot c_{t-1} + i_t \cdot l_t \\ h_t &= o_t \cdot \tanh(c_t) \end{aligned} \tag{1}$$

where W_i, W_f, W_o and W_l have dimensions D×2N, D is the number of memory cells and N is the dimension of the input vector. These matrices represent the parameters of the network. LSTM is local in space and time since its computational complexity per time step and weight is $O(1)$ [12].

B. Computational procedure

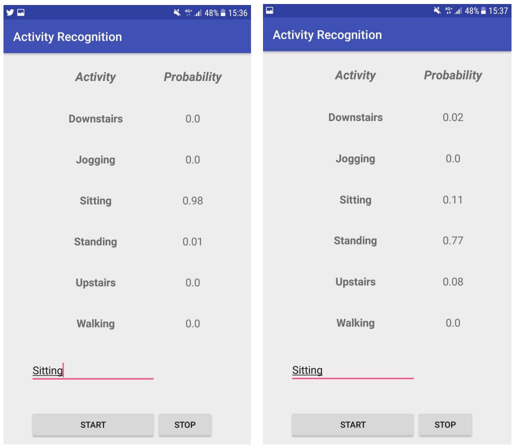
The computation pipeline used in this study follows standard procedure. A schematic flowchart in Fig. 1 shows an outline of this process. It begins with raw data, collected in controlled laboratory setting, which is processed into sequences of length 200. Afterwards, the generated inputs are divided into a training (80%) and testing (20%) datasets. The training set is used to train the LSTM network and to generate the model. The model alongside the testing set is used to calculate the accuracy of the algorithm. Additionally, the model is transferred to an Android device as part of a mobile application which performs accelerometer measurements, real time prediction and calculation of the LSTM accuracy.

C. Implementation tools

For the implementation of the LSTM network, the Python library TensorFlow was used [16]. The data was divided into 10-second segments (sequences of 200 samples) and each segment was used as an input in the network. We use three LSTM layers with 64 neurons each. Additionally, we use a L2 regularization with loss of 0.0015. Recently, the Adam Optimizer has gained a lot of popularity, so we decided to use this optimizer with a learning rate of 0.0025. The neural network was trained for 100 epochs, as we observed that longer training than 100 epochs doesn't improve the performance.

D. Real time Android application

In order to test the real-time performance of our model, we developed an Android application. The application collects measurement from the device's accelerometer every 50ms and outputs the probability of each of the six activities occurring during the previous 10-second window. For the implementation of this application we exported our TensorFlow model and imported it in Android using TensorFlowInferenceInterface. Additionally, we use the text-to-speech Android API which tells the user the predicted activity during the previous 10-second window. A view of this application is given in Fig. 2.



Activity Recognition

Activity	Probability
Downstairs	0.0
Jogging	0.0
Sitting	0.98
Standing	0.01
Upstairs	0.0
Walking	0.0

Sitting

Activity Recognition

Activity	Probability
Downstairs	0.02
Jogging	0.0
Sitting	0.11
Standing	0.77
Upstairs	0.08
Walking	0.0

Sitting

Fig.2 Interface of the Android mobile application

When algorithms are needed to be tested under field conditions, the standard procedure is by using diary, where users record their activities. This is labor-intensive task prone to errors. Therefore, our mobile application can operate in a testing mode for users that want to participate in the process of accuracy identification. As can be seen in Fig. 2, in this mode, the user can manually enter the perfromed activity, so the application is able to calculate the accuracy for each user separately.

III. Data collection and computation

In this section, the datasets used in our research are explained in detail.

A. *Lab Data*

Our LSTM based algorithm was trained and evaluated on data collected in controlled laboratory setting as described in [17]. Hereafter, we refer to this data as *"Lab Data"*. The data was collected from 29 volunteers carrying a smartphone in their front leg pocket. The subjects were asked to do six specific activities: sitting, standing, walking, jogging, ascend stairs and descend stairs. The accelerometer data was collected using an Android application. A sample was collected every 50ms. Every sample contains a timestamp, user ID, as well as the x, y and z accelerometer values.

B. *Field data*

To test the generalization power of our algorithm, we collected our own dataset under field conditions, doing the same six activities outdoors in a less controlled environment. Hereafter, we refer to this data as *"Field Data"*. The accelerometer data was collected from two subjects, one male and one female, carrying a smartphone in front leg pocket. Our Android application contains the same fields and records the data with the same frequency as in [17]. We plot 10-second windows of the accelerometer data for all activities in Figures 3-8. We observe that "Sitting" and "Standing" do not have periodic behavior but can be distinguished based on the relative magnitudes of the x, y and z values. For all other activities we observe periodic behavior. As expected, the "Jogging" activity shows greatest acceleration, followed by "Walking", while "Upstairs" and "Downstairs" having smaller acceleration.

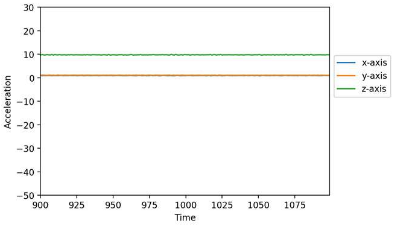


Fig. 3 Time series for activity "Sitting"

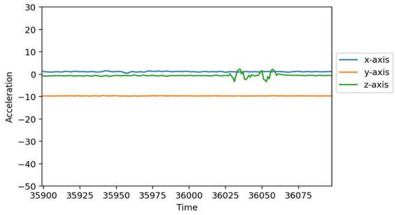


Fig. 4 Time series for activity "Standing"

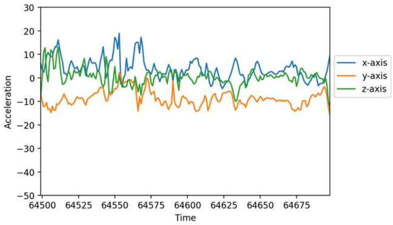


Fig. 5 Time series for activity "Walking"

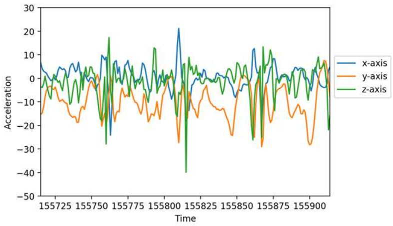


Fig 6. Time series for activity "Jogging"

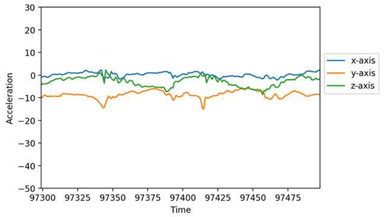


Fig 7. Time series for activity "Upstairs"

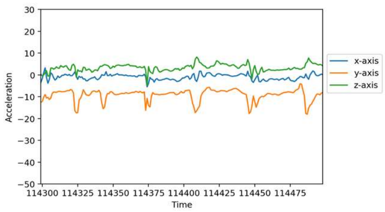


Fig 8. Time series for activity "Downstairs"

IV. RESULTS AND DISCUSSION

The results of our LSTM based algorithm were compared with the results from other methods reported in [17], as given in Table I. From the results, we can conclude that LSTM gives almost as good overall performance as the best method used in [17]. The advantage of LSTM is that it works directly with the raw accelerometer data, and completely bypasses the process of generating hand-crafted features. This allows the network to better learn the underlying data distribution. Furthermore, for the "Walking", "Upstairs", "Sitting" and "Standing" classes, LSTM gives better performance than other methods investigated in [17], i.e. decision trees J48, logistic regression, multilayer perceptron and straw man. The straw man strategy in [17] always predicts the most frequently occurring activity. LSTM achieves perfect performance on the "Standing" class. Additionally, for the classes of "Jogging" and "Downstairs", the LSTM approach is only slightly worse than the best achieved performance in [17]. Therefore, LSTM works almost as good or better than the approaches that use hand-crafted features which means that we can confidently skip the process of features selection and work directly with the raw accelerometer data.

TABLE I. % OF CORRECTLY PREDICTED ACTIVITIES

	J48	LOGISTIC REGRESSION	MULTILAYER PERCEPTRON	STRAW MAN	LSTM
WALKING	89.90	93.60	91.70	37.20	**95.30**
JOGGING	96.50	98.00	**98.30**	29.20	96.50
UPSTAIRS	59.30	27.50	61.50	12.20	**67.00**
DOWNSTAIRS	**55.50**	12.30	44.30	10.00	50.30
SITTING	95.70	92.20	95.00	6.40	**96.90**
STANDING	93.30	87.00	91.90	5.00	**100.00**
OVERALL	85.10	78.10	**91.70**	37.20	88.60

In Table II we present the confusion matrix for the LSTM model. The most important activities to analyze are the "Upstairs" and "Downstairs" activities which are most difficult to recognize. From the confusion matrix we can see that these two activities are mostly confused with each other, and less commonly with the "Jogging" activity. Looking at Fig. 7 and Fig. 8, it can be seen that these two activities are very similar and thus all the algorithms are facing difficulties to distinguish them.

The comparison between the performance on the *"Lab Data"* [17] and the *"Field Data"* is presented in Table III. From the results it can be seen that LSTM algorithm performs few percentage points worse on the *"Field Data"* for the classes of "Walking", "Jogging" and "Standing". We manage to achieve slightly better performance for the classes of "Upstairs" and "Sitting". On the other hand, we observe much worse performance for the "Downstairs" activity. We can conclude that the "Downstairs" activity is much harder to be predicted compared to other activities. This conclusion is further supported by the fact that even on the *"Lab Data"* no algorithm can achieve accuracy greater than 55% for the "Downstairs" activity. Finally, we can conclude that our algorithm is able to generalize well on data from different subjects and sources and still achieve comparable performance.

TABLE II. CONFUSION MATRIX FOR LSTM

Predicted	**Downstairs**	**Jogging**	**Sitting**	**Standing**	**Upstairs**	**Walking**
Actual						
Downstairs	75	9	0	0	55	10
Jogging	8	498	0	0	5	5
Sitting	0	0	95	2	1	0
Standing	0	0	0	70	0	0
Upstairs	33	18	1	0	128	11
Walking	8	7	2	0	12	595

TABLE III. COMPARISON OF CORRECTLY PREDICTED ACTIVITIES FOR LAB DATA AND FIELD DATA (IN %)

	LAB DATA	FIELD DATA
WALKING	**95.30**	90.49
JOGGING	**96.50**	96.11
UPSTAIRS	67.00	**75.69**
DOWNSTAIRS	**50.30**	21.02
SITTING	96.90	**97.84**
STANDING	**100.00**	97.20
OVERALL	**88.60**	82.20

Each of the Figures 9-14 correspond to one activity. Each plot shows the predicted activity from *"Field Data"*. From the results, we can conclude that "Sitting" and "Standing" (Fig. 9 and Fig. 10) are correctly predicted except for several samples which can be considered as random noise. "Walking" is mainly confused with "Jogging" (Fig. 11). During our testing we observed that faster walking is generally classified as "Walking", which is probably caused by the collection strategy of the original dataset (*"Lab Data"*) on which our algorithm is trained. The "Jogging" is much less frequently confused with "Walking" which can be explained with a user slowing down to rest during the testing stage (Fig. 12). The "Upstairs" activity is mostly confused with "Walking", which might be caused by the difference in the type of stairs or the speed of climbing between the *"Lab Data"* and *"Field Data"* datasets (Fig. 13). "Downstairs" is most often confused with "Upstairs", a trend that is observed even in the original *"Lab Data"* dataset (Fig. 14).

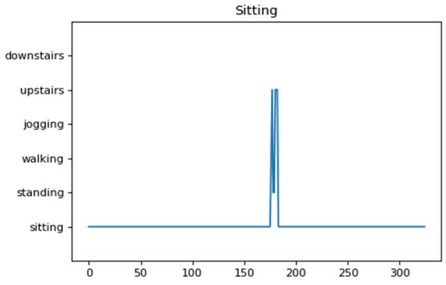


Fig. 9 Activity prediction for "Sitting" activity

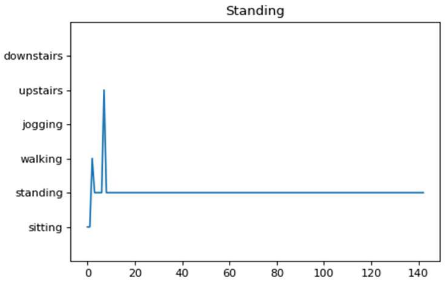


Fig. 10 Activity prediction for “Standing” activity

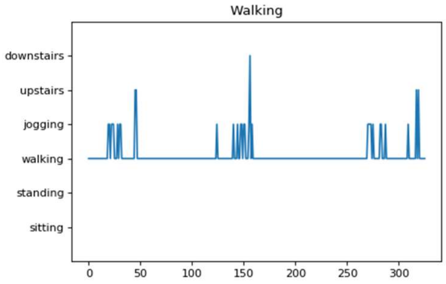


Fig. 11 Activity prediction for “Walking” activity

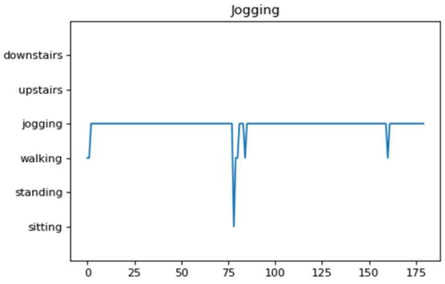


Fig. 12 Activity prediction for “Jogging” activity

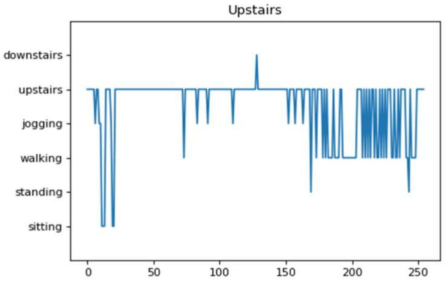


Fig. 13 Activity prediction for “Upstairs” activity

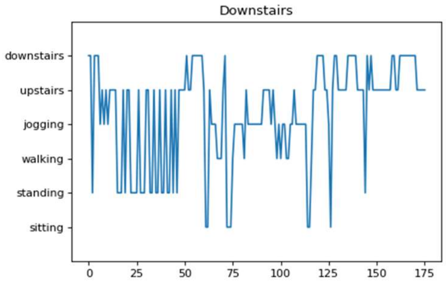


Fig. 14 Activity prediction for “Downstairs” activity

V. CONCLUSION

In this paper we developed and implemented a lightweight algorithm for human activity recognition from smartphone accelerometer data. It is based on long short term memory (LSTM) network, a relatively new approach suitable for multimodal wearables. LSTM does not require expert knowledge in designing features and learns features from raw accelerometer data, completely bypassing the process of generating hand-crafted features. Therefore, LSTM is lightweight with computational complexity of $O(1)$, appropriate to operate in real-time on wearables and smartphones.

LSTM was compared with other approaches from the literature [17], i.e. decision trees J48, logistic regression and multilayer perceptron, and it was shown that LSTM works almost as good or better than the approaches that use hand-crafted features.

To evaluate the usability of LSTM for real smartphone applications, we trained and tested our algorithm on data collected in controlled setting, but we also tested on data collected under field conditions. Our initial results show that LSTM algorithm performs almost equally good on smartphone data collected under field conditions, which makes it (and its future improvements) suitable candidate to be improved and implemented for commercial mobile applications.

ACKNOWLEDGMENT

This work was partially financed by the Faculty of Computer Science and Engineering, University “Ss. Cyril and Methodius”, Skopje, Macedonia.

REFERENCES

[1] Zdravevski, Eftim, Biljana Risteska Stojkoska, Marie Standl, and Holger Schulz. "Automatic machine-learning based identification of jogging periods from accelerometer measurements of adolescents under field conditions." *PloS one* 12, no. 9 (2017): e0184216.

[2] Robusto KM, Trost SG. Comparison of three generations of ActiGraph™ activity monitors in children and adolescents. Journal of Sports Sciences. 2012;30(13):1429–1435.

[3] Fortino, Giancarlo, Stefano Galzarano, Raffaele Gravina, and Wenfeng Li. "A framework for collaborative computing and multi-sensor data fusion in body sensor networks." *Information Fusion* 22 (2015): 50-70.

[4] Su, Xing, Hanghang Tong, and Ping Ji. "Activity recognition with smartphone sensors." *Tsinghua Science and Technology*19, no. 3 (2014): 235-249.

[5] Bayat, Akram, Marc Pomplun, and Duc A. Tran. "A study on human activity recognition using accelerometer data from smartphones." *Procedia Computer Science* 34 (2014): 450-457.

[6] Khan, Adil Mehmood, Y-K. Lee, S. Y. Lee, and T-S. Kim. "Human activity recognition via an accelerometer-enabled-smartphone using kernel discriminant analysis." In *Future Information Technology (FutureTech), 2010 5th International Conference on*, pp. 1-6. IEEE, 2010.

[7] Anguita, Davide, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge L. Reyes-Ortiz. "Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine." In *International workshop on ambient assisted living*, pp. 216-223. Springer, Berlin, Heidelberg, 2012.

[8] Lu, Yonggang, Ye Wei, Li Liu, Jun Zhong, Letian Sun, and Ye Liu. "Towards unsupervised physical activity recognition using smartphone accelerometers." *Multimedia Tools and Applications* 76, no. 8 (2017): 10701-10719.

[9] Oneto, Luca, Jorge LR Ortiz, and Davide Anguita. "Constraint-Aware Data Analysis on Mobile Devices: An Application to Human Activity Recognition on Smartphones." In *Adaptive Mobile Computing*, pp. 127-149. 2017.

[10] Ordóñez, Francisco Javier, and Daniel Roggen. "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition." *Sensors* 16, no. 1 (2016): 115.

[11] Sainath, Tara N., Oriol Vinyals, Andrew Senior, and Haşim Sak. "Convolutional, long short-term memory, fully connected deep neural networks." In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pp. 4580-4584. IEEE, 2015.

[12] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." *Neural computation* 9, no. 8 (1997): 1735-1780.

[13] Felix A. Gers; Jürgen Schmidhuber; Fred Cummins (2000). "Learning to Forget: Continual Prediction with LSTM". *Neural Computation*. **12** (10): 2451–2471.

[14] McGraw, Ian, Rohit Prabhavalkar, Raziel Alvarez, Montse Gonzalez Arenas, Kanishka Rao, David Rybach, Ouais Alsharif et al. "Personalized speech recognition on mobile devices." In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pp. 5955-5959. IEEE, 2016.

[15] Alsharif, Ouais, Tom Ouyang, Françoise Beaufays, Shumin Zhai, Thomas Breuel, and Johan Schalkwyk. "Long short term memory neural network for keyboard gesture decoding." In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pp. 2076-2080. IEEE, 2015.

[16] Abadi, Martín, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin et al. "TensorFlow: A System for Large-Scale Machine Learning." In *OSDI*, vol. 16, pp. 265-283. 2016.

[17] Jennifer R Kwapisz, Gary M Weiss, and Samuel A Moore. Activity recognition using cell phone accelerometers. ACM SigKDD Explorations Newsletter, 12(2):74–82, 2011.