Human Activity Recognition With RNN

Shailendra Bhandari
Department of Computer Science
OsloMet- Oslo Metropolitan University
Oslo, Norway
s366261@oslomet.no

Divya Annamalai
Department of Computer Science
OsloMet- Oslo Metropolitan University
Oslo, Norway
s366265@oslomet.no

***Abstract*—In this project, we present a convolutional neural network (CNN) and long short term memory (LSTM) classifiers for human activity recognition. Both the CNN and the LSTM networks have been extensively researched deep learning models in the past. But they have been studied in isolation except, only few attempts have been made to use these methods for activity recognition. Since last few years these approach have been extensively used by researchers with higher accuracy. Researchers are striving for the system whose ability is to use as few resources as possible to recognize a human activity from a raw sensor data. The deep neural network model we used can extract the activity features automatically and classify them with as few model parameters as possible. We used LSTM, CNN and CNN+LSTM deep learning based activity recognition architecture. The CNN+LSTM approach improves the predictive accuracy of human activities from the raw sensor data. The achieved overall accuracy of CNN+LSTM model is 92.37%. Similarly, the accuracy of single layer LSTM, multi-layer LSTM, and 1D-CNN model are 90.91%, 92.05% and 91.52%, respectively on the UCI HAR public dataset. We also compared its performance against few other recent research. It competes favorably against deep neural network architecture that have been proposed and implemented in the past.**

***Index Terms*—Recurrent Neural Network (RNN), Human Activity Recognition (HAR), Long Short Term Memory (LSTM), Convolutional neural network (CNN)**

I. Introduction

Human activity recognition plays an important role in people's daily lives for its ability to compute the advanced knowledge about human activity from the raw sensor data [1]. With the advancement of computer vision, the technology of HAR has become a popular research direction for indoor and outdoor activities. Some of the successful HAR applications include in-home activities of human users (particularly elderly people) with wrist-worn device sensing [2], video surveillance: action similarity labeling (ASLAN), which aims at verifying whether a pair of videos contain the same type of action or not [3], explore deep, convolutional, and recurrent approaches across three representative datasets that contain movement data captured with wearable sensors [4]. With the emergence of deep learning in the field of HAR, the task of human activity recognition has become straightforward. Machine (ML) and deep (DL) learning models are readily made available by frameworks like TensorFlow, PyTorch, Scikit, and others not to mention the Keras API [1] that has made it easy to build and experiment with models.

With the rapid development of sensor technology and computing technology, the popularity of the sensor-based HAR has become more and more popular and is widely used with privacy being well protected. In this HAR dataset and usually in a typical HAR scenario, a user with a sensor device is equipped with gyroscope and accelerometer sensors that continuously send sensor data to a server that enables continuous monitoring of the activity of the user. Using the learning models, it has become a lot easier to train a model to recognize the specific activities from the raw sensor data efficiently and conveniently. The popular machine learning models are SVM, k-nearest classifier, logistic regression, decision tree, and so on. These approaches managed to reach impressive classification and recognition of accuracies. However, feature extraction is essential in using these machine learning classifiers. To get rid of this, the deep learning approach has come into play where we can feed raw data and efficiently make predictions. Some of the popular approaches that have been proposed and used are CNN [5] which are spatially deep, and the LSTM which is temporally deep.

In this project report, we present different CNN and LSTM deep neural models for human activity recognition. We use a hybrid, the CNN+LSTM model which was researched in the past. Our work seeks to leverage the strength of consolidating these two models regarding human activity recognition. The paper is structured as follows: Section II discusses the importance of the deep learning model in HAR, and section III highlights the recent research work. In section IV we discuss our methods and implementation (Experimental setup). Section V and VI discuss the results and conclusion of this project work.

II. Why a deep learning model for HAR

Machine learning methods have been the core of research into human activity recognition since deep learning methods could automatically extract appropriate features from the raw sensor data during the training phase and process them to get appropriate outputs. The deep learning method has already been successful in image classification, voice recognition,

[1] https://keras.io/

pattern recognition, and natural language processing; it is a new research direction in human activity pattern recognition [6]–[8]. However, there are several drawbacks to conventional pattern recognition methods.

Convolution neural networks (CNNs) leverage three main ideas: sparse interaction, parameter sharing an equivalent representation [9]. CNN is a type of deep neural network which uses convolution operations to learn kernels for collecting features from input data. Different topological models can be used to define such kernels.

- 1D kernels, which are mainly used for temporal processing using a defined window [10];
- 2D kernels, which are mainly used for spatial relation learning.

A convolutional network's first component is the convolutional layer. While additional convolutional or pooling layers can be implemented after convolutional layers, the fully-connected layer is the last. The CNN becomes much more complex with each layer, identifying more significant image parts. Earlier layers concentrate on essential elements like colors and borders. As the visual data travels through the CNN layers, it distinguishes more significant elements or forms of the item, subsequently identifying the target object.

1) Convolutional Layer: The convolutional layer is the most critical feature of a CNN because most of the computation occurs. It requires input data, a filter, and a feature map, amongst many other things. Let us pretend the input is a colour image of a 3D matrix of pixels. This means that the input will have three dimensions: height, width, and depth, which correspond to a picture's RGB colour space. A feature detector, also known as a kernel or a filter, will check for the presence of the feature across the image's receptive fields. This method is known as convolution.

A two-dimensional (2-D) weighted array representing a piece of the image is used as the feature detector. The filter size, which can vary, usually is a 3x3 matrix, which influences the size of the receptive field as well. The filter is then applied to a section of the image, and the dot product of the input pixels and the filter is determined. The output array receives this dot product. The filter then shifts by one stride, and the procedure is repeated until the kernel has swept across the entire image. A feature map, activation map, or convoluted feature is the ultimate output from a series of dot products generated by the input and the filter.

Before the neural network training begins, three hyper parameters determine the output volume size that must be established. These are some of them:

1) The output depth is affected by the number of filters used.
2) The kernel's stride is the number of pixels it traverses across the input matrix.
3) When the filters do not fit the input image, zero-padding is commonly utilized.

2) Pooling Layer: Down sampling, also known as pooling layers, is a dimensionality reduction technique used to reduce the number of factors in the input. The pooling process spreads a filter across the whole input, similar to the convolutional layer; however, this filter does not have any weights. Instead, the kernel uses a clustering method to populate the output array from the values in the receptive field. Pooling can be divided into two categories:

Max pooling: The filter takes the pixel with the highest value to transmit to the output array as it goes across the input. In comparison to average pooling, this strategy is employed more frequently.

Average pooling: The filter measures the average value inside the receptive field as it passes across the input and transmits it to the output array. While the pooling layer loses much information, it does have a few advantages for CNN. They assist in reducing complexity, increasing efficiency, and reducing the risk of overfitting.

3) Fully-Connected Layer: In partially linked layers, the pixel values of the input image are not directly connected to the external layer. Each node in the output layer, on the other hand, directly connects to a node in the preceding layer in the fully-connected layer.

4) Long-Short Term Memory (LSTM): The long short-term memory (LSTM) architecture is a deep learning architecture that involves a recurrent neural network (RNN) published [11] in 1997 by Sepp Hochreiter and Jürgen Schmidhuber. Unlike regular feed-forward neural networks, LSTM has feedback connections. Therefore, it is well suited to learn from experience to classify, process, and predict time series when they are very long time lags of unknown size between important events. A memory cell in LSTM is composed of four main elements, an input gate, a neuron with a self recurrent connection, a forget gate, and an output gate. An input gate can allow the incoming signal to alter the state of the memory cell or block it. The reconnect connection ensures barring any outside inference where the state of the memory cell can remain constant from one-time step to another. The forget gate can modulate the memory cell's self-recurrent connection, allowing the cell to remember or forget its previous state, as needed. And finally, the output gate can allow the state of the memory cell to have an effect on other neurons or prevent it.

III. Related Works

The major goal of human activity recognition is to notice the daily behavior of the people through the analysis of the observation obtained and their neighboring environments of living [12]. Various research works in the field of human activity recognition using deep learning and general classification methods have been done. Machado et.al. [13] mention the K-means clustering algorithm-based unsupervised machine learning method was used to identify human activities. The K-means methodology presented promising accuracy results for person-dependent and independent cases, with 99.29% and 88.57%, respectively. It means that the performance of the methodology is limited when there is a mixture of static and dynamic activity in the dataset. Similarly, Jiang et.al. in their research work extracted motion, audio, and spatial features using CNN. They used the LSTM model on appearance and short-time information and ignores long temporal dynamics in the video. Since the LSTM model is found to be more advantageous than the CNN models [14], researchers have proposed certain techniques for LSTM-based HAR models. Similarly, Zhao et. al. [15] proposed a bidirectional LSTM architecture with the advantage to concatenate the forward state and backward state i.e. positive and negative time direction. The accuracy of bidir-LSTM on the public domain UCI dataset is 91.1% [15]. Wang et.al. [16] used a hybrid model (CNN and LSTM) which accurately recognizes activities and their transitions on a wearable sensor-based data set. The accuracy of this model is 95.87%. The CNN learns local features from the original sensor data, and LSTM extracts time-dependent relationships from local features and realizes the fusion of local features and global features, a fine description of basic and transition movements to accurately identify the two motion patterns.

IV. Experimental setup

A. Dataset

The human activity recognition (HAR) dataset was introduced by Davide Anguita et al. [17] and is available in the public domain http://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones. Similarly, the information of the data set can be found in detail in the same URL. The data was collected using three different sensors; accelerometer, gyroscope, and magnetometer that are built into IMU devices, and the smartphones to recognize the activity being performed by the user of the device. The HAR dataset has 6 activities with 30 volunteers with an age bracket of 19-48 years. The activities are walking, walking upstairs, walking downstairs sitting standing, and lying. Each of these activities consists of 3D raw signals extracted from above mentioned 3 sensors [17]. There are 7352 training 2947 test samples in the dataset. This dataset also includes postural transitions that occur between the static postures: standing to sitting, sitting to standing, sitting to laying, laying to sitting, standing to lay, laying to standing

B. Data Visualization and preprocessing

The signals samples were pre-processed by applying noise filters and sampled in fixed-width sliding windows of 2.56 sec with a 50 percent overlap (i.e. 128 readings/windows). The sensor acceleration signal, which has gravitational and body motion components, was separated using a Butterworth low-pass filter into body acceleration and gravity. The activities of UCI-HAR data set with activities per data-points is shown in Figure [1].

Out of six activities, three activities (sitting, standing and lying) are the stationary activities whereas the rest of three (walking, walking upstairs and walking down) are the moving activities. The stationary and the moving activities are shown visually in Figure [2].

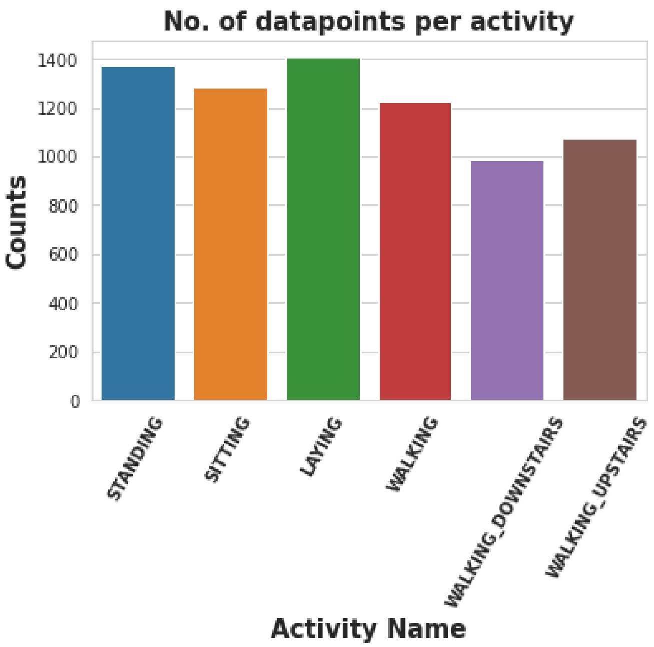


Fig. 1. The number of data points per activity of HAR dataset.

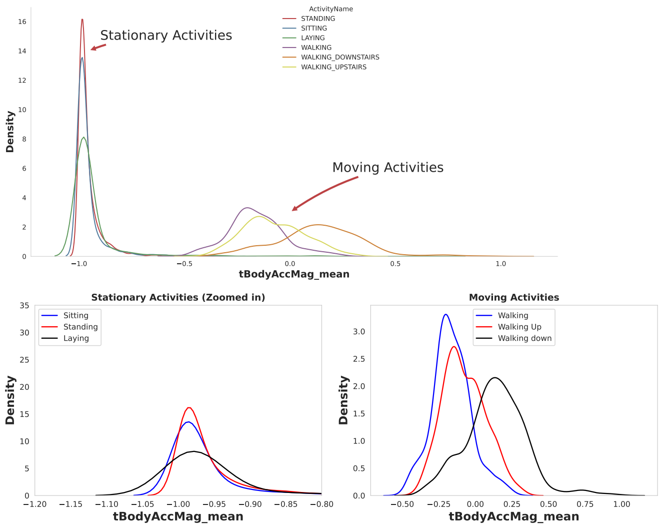


Fig. 2. The stationary and moving activities of six daily life activities of HAR dataset.

C. Evaluation Protocol

In this project we are using LSTM recurrent neural network and CNN-LSTM model. Figure [3] shows the network architecture of LSTM model to predict and classify six different activities with maximum accuracy. This model is implemented

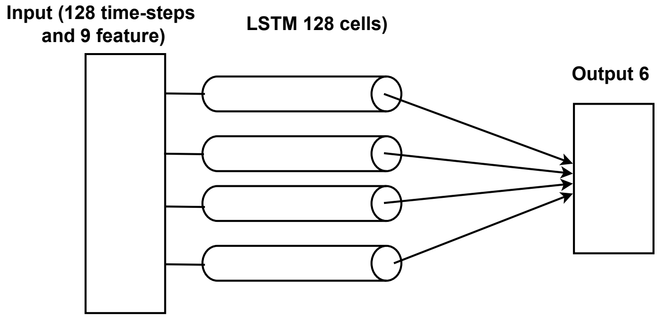


Fig. 3. The network architecture of LSTM model.

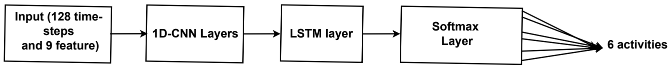


Fig. 4. The network architecture of CNN LSTM model.

using Keras API sequential model with a TensorFlow backend. The model was trained in a fully supervised manner and the gradient was back-propagated from the Softmax layer to the LSTM layer. We used categorical cross-entropy to evaluate multi-class classification. These are tasks where an example can only belong to one out of many possible categories, and the model must decide which one. We used Adam optimizer as stochastic optimization algorithm.

Similarly, Figure [4] shows a network architecture of another model we are implementing in this project. It consists of 10 different layers. We used 1D convolution layer (tf.keras.layers.Conv1D) with 64 filters and kernel_size =3 which specifies the length of the 1D convolution window. This layer used the ReLU activation function with 128-time steps, 64 filters, and 9 features. After, we added a maxPooling1D with pool_size of 2 followed by Conv1D with kernel_size 3 and activation function ReLU. Next, we used a flatten layer whose job is to format the feature data from this section to be consumed by the LSTM layer in the next layer. The next layer is the LSTM layer with 128 hidden neurons followed by dropout and flatten layers. Finally, we implemented a dense layer with 6 activities of daily life with the Softmax classifier.

V. Results

A. *Standard Machine Learning Classifiers*

Typical machine learning algorithms such as support vector machine (SVM), decision tree (DT), and Logistic Regression (LR) are used in the classification and recognition of human activities. We use three machine learning classifiers and are compared them with each other to obtain the best accuracy rate among the classifiers. The evaluation technique we used for these classifiers is the confusion matrix. The confusion matrix offers a complete overview by summarizing the classification results. The accuracy of the three different classifiers is tabulated in the Table [I]. Among the three classifiers, Linear SVC has the highest accuracy whereas DT has minimum accuracy. Zaki et.al. [18], in their paper, mentioned that the highest accuracy obtained was for LR with 96.20%. Also, in the results of Anguita et.al. [17] the accuracy of the SVC model is 96%.

TABLE I
The results from the three different machine learning classifiers.

ML classifiers	Accuracy	Error
Logistic Regression	95.83%	4.174%
Linear SVC	96.67%	3.325%
Decision Tree	87.17%	12.83%

B. *Deep neural network*

To evaluate our model and see the performance, we ran the experiment thoroughly and thus obtained results will be shown in this section. Figure [6] shows the accuracy attained from the various models we trained on this work with 6 classes. We can see from the figure that sitting and standing have the lowest performance for all three models whereas rest of the activities have almost similar accuracy. Also, we cannot compare which model is performing best for all the activities.

Table [II] shows the overall performances of all four models we used. We can see that the 1D-CNN LSTM model outperforms all other models with an accuracy of 92.37%. This result is followed by the multi-layer LSTM with an accuracy of 92.05% accuracy. Similarly, Figure [7] shows the calculated accuracy, F1-score, precision, and recall scores for all the models we used in the experiment. It is important to note that since we have used the categorical cross-entropy loss in the model compiler and Softmax activation in the output layers as the data set is multivariate, we have to calculate the how accurately the model has evaluated a sample in the right class by associating a probability to each output class. This can be done by calculating Softmax loss, which should have to be done in the future. Figure [5] show the confusion matrix for three models, multi-layer LSTM, 1D-CNN and 1D-CNN + LSTM.

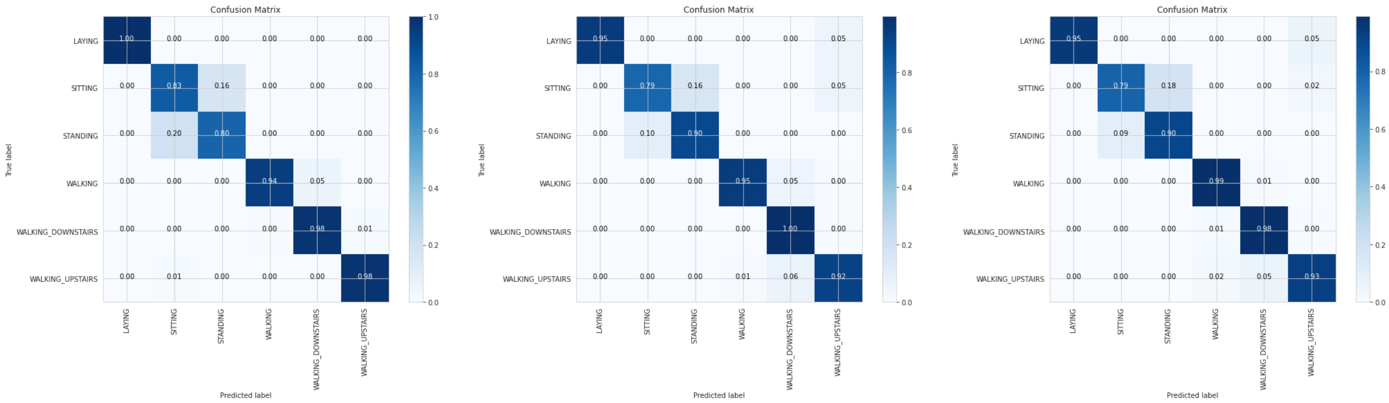
TABLE II
Performance of the model used for multi-class output human activity recognition.

Machine learning model	Accuracy	f1-scores	Precision	Recall
Single layer LSTM	90.91%	90.81%	91.19%	90.47%
Multi layer LSTM	92.05%	92.15%	92.17%	92.14%
1D-CNN	91.52%	91.59%	91.66 %	91.52%
1D-CNN + LSTM	92.37%	92.44%	92.44%	92.44%

The results above to some extent show that the hybrid CNN + LSTM model outperforms the rest of all models with higher accuracy. Since the data set is time-series data, it is expected that the LSTM model performs better than the CNN model. However, the 1D-CNN also has good accuracy. To doubt this conclusion, hyper-tuning parameters are essential for calculating the learning rates and to fine tune the models.

C. *Comparison with recent work*

Ronal et.al. [19] used CNN_LSTM, CNN_LSTM Dense LSTM, and LSTM Dense model on the same HAR data set.



a. Multi-layer LSTM

b. 1D CNN

c. 1D CNN +LSTM

Fig. 5. Confusion matrix showing the accuracy of the model for each of 6 activities of daily life for all the models.

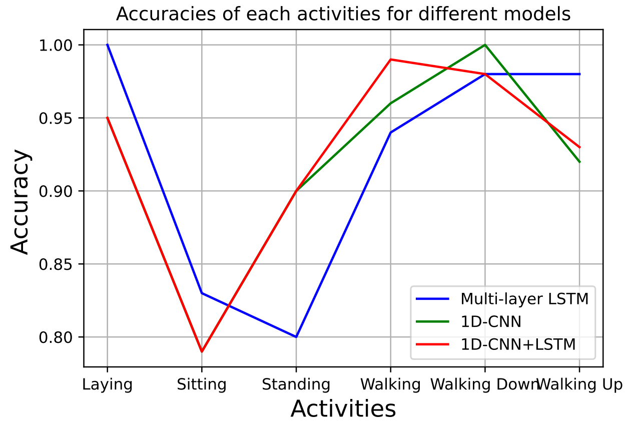


Fig. 6. Line graph sh owning the accuracy of different models we used for different activities of daily life.

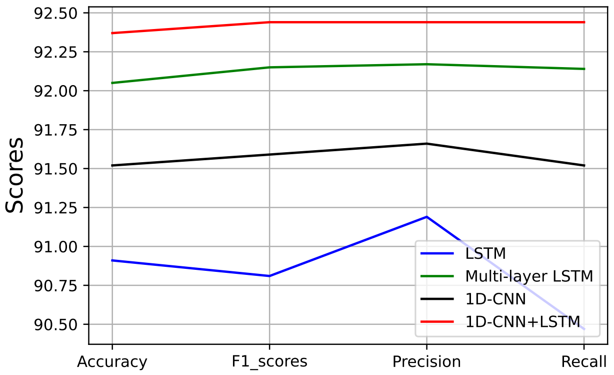


Fig. 7. Line graph showing the accuracy, precision, f1-score and recall for all the models we used in this experiment.

In their work, the calculated accuracy is 92.13%, 91.55%, 91.28%, and 91.40% for CNN_LSTM, CNN_LSTM Dense LSTM, and LSTM Dense models, respectively. The network architecture of their CNN_LSTM Dense, to some extent, is the same as ours except for a different number of neurons in the hidden layers of LSTM. Overall, we can see the improvement in the accuracy in our models in comparison to the work of Ronal et.al [19].

VI. Conclusions

In this project, we used four different types of deep neural networks as well as three different machine learning classifiers to calculate and evaluate human activities. The CNN + LSTM model to human activity recognition seeks to improve the accuracy of the activity recognition by leveraging the robustness in feature extraction of a CNN network while taking advantage of the work an LSTM model does for time series forecasting and classification. As the CNN LSTM models feature both spatially and temporally performed compared with another deep neural network, it is proven with increased accuracy when computing with the raw signal data as input. As mentioned in the result section, to further and properly evaluate these models, we need to calculate the learning rate by hyper-tuning the parameters of the models. Also, the model can be hyper-tuned by properly selecting the batch size and regularization of the model. Because of the time limitation, this task is not done here and can be done in the future.

The data was collected for the experimental purpose from healthy volunteers. However, testing this model in real-world applications might be challenging. Therefore, future consequences of the real-world application of these methods should have to address.

VII. Contributions

Equal contribution by both of us on codes as well as on report. Similarly we have equal contribution on short project.

VIII. Appendix

A. *Implemented code*

The above mentioned method is implemented using Tensorflow 2.8.0, sklearn 1.0.1, keras 2.8.0 and Python version 3.8 and was run in DELL inspiration 15 7000 GPU-

NVidia GeForce graphics. The code is made available public in GitHub in the link https://github.com/Shailendra995/Acit4530_final_project.git.

B. Network models used

As mention earlier, we used four different deep learning models in the experiments. The model was trained in a fully supervised manner to back propagate the gradient from the Softmax layer to the input layer. Network parameters are optimized by using mini-batch gradient descent method. The optimizer used is Adam optimizer. Adam optimizer is widely used due to its advantages in simple implementation, efficient calculation, and low memory demand. Figure [8] shows the network architecture of single layer LSTM, Figure [9] shows the network architecture of 1D-CNN + LSTM model, and Figure [10, 11] shows the network model for multi-layer LSTM and CNN model, respectively. Also the accuracy plot for both training and the test data for all the model we used in this project are shown in Figure [12].

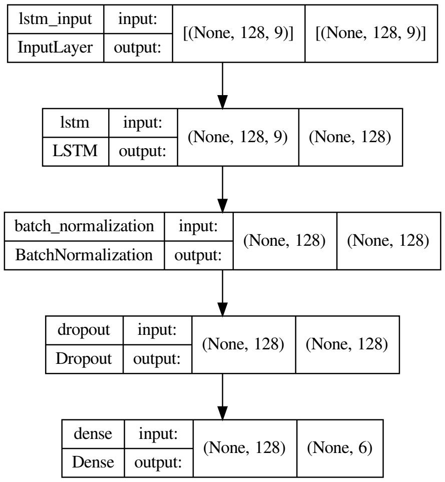

lstm_input | InputLayer | input: | output: | [(None, 128, 9)] | [(None, 128, 9)]
lstm | LSTM | input: | output: | (None, 128, 9) | (None, 128)
batch_normalization | BatchNormalization | input: | output: | (None, 128) | (None, 128)
dropout | Dropout | input: | output: | (None, 128) | (None, 128)
dense | Dense | input: | output: | (None, 128) | (None, 6)

Fig. 8. Single layer LSTM network architecture.

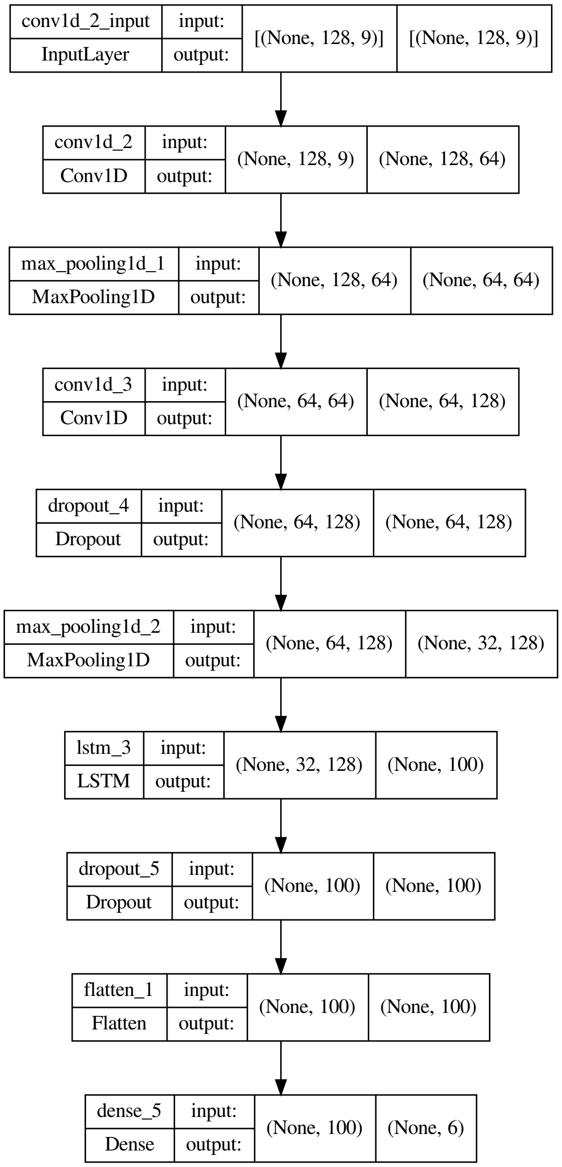

conv1d_2_input | InputLayer | input: | output: | [(None, 128, 9)] | [(None, 128, 9)]
conv1d_2 | Conv1D | input: | output: | (None, 128, 9) | (None, 128, 64)
max_pooling1d_1 | MaxPooling1D | input: | output: | (None, 128, 64) | (None, 64, 64)
conv1d_3 | Conv1D | input: | output: | (None, 64, 64) | (None, 64, 128)
dropout_4 | Dropout | input: | output: | (None, 64, 128) | (None, 64, 128)
max_pooling1d_2 | MaxPooling1D | input: | output: | (None, 64, 128) | (None, 32, 128)
lstm_3 | LSTM | input: | output: | (None, 32, 128) | (None, 100)
dropout_5 | Dropout | input: | output: | (None, 100) | (None, 100)
flatten_1 | Flatten | input: | output: | (None, 100) | (None, 100)
dense_5 | Dense | input: | output: | (None, 100) | (None, 6)

Fig. 9. 1D-CNN + LSTM network architecture.

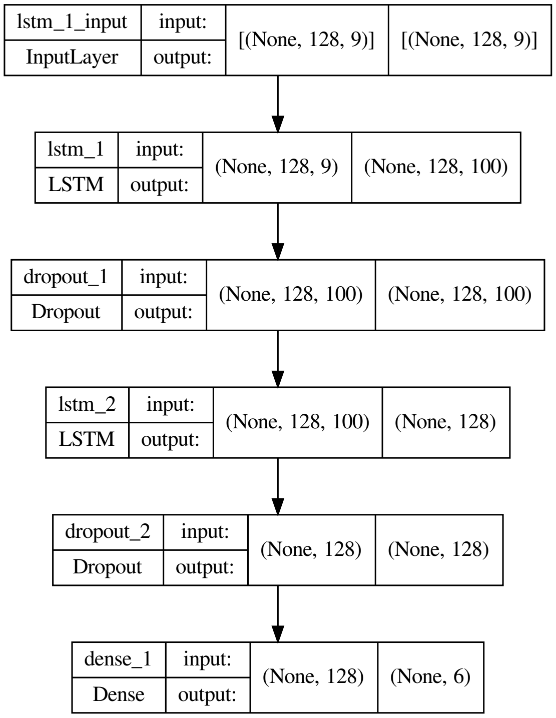

lstm_1_input | input: | [(None, 128, 9)] | [(None, 128, 9)]
InputLayer | output:
lstm_1 | input: | (None, 128, 9) | (None, 128, 100)
LSTM | output:
dropout_1 | input: | (None, 128, 100) | (None, 128, 100)
Dropout | output:
lstm_2 | input: | (None, 128, 100) | (None, 128)
LSTM | output:
dropout_2 | input: | (None, 128) | (None, 128)
Dropout | output:
dense_1 | input: | (None, 128) | (None, 6)
Dense | output:

Fig. 10. Multi-layer LSTM network architecture.

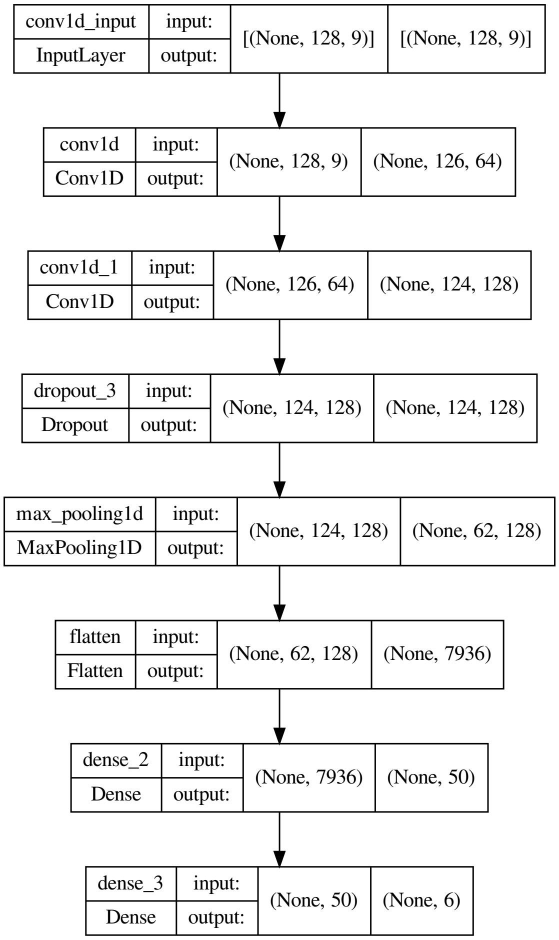
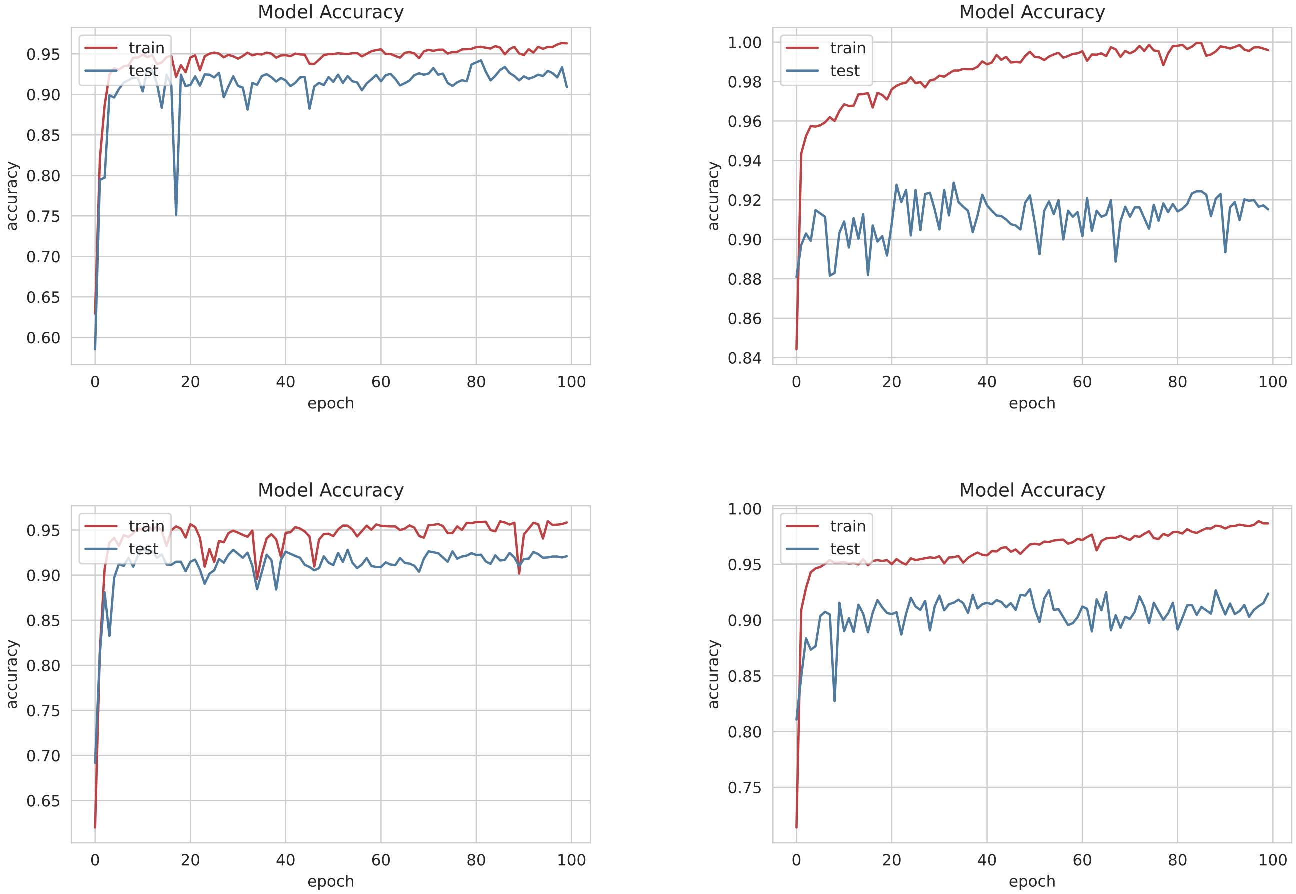

conv1d_input | input: | [(None, 128, 9)] | [(None, 128, 9)]
InputLayer | output:
conv1d | input: | (None, 128, 9) | (None, 126, 64)
Conv1D | output:
conv1d_1 | input: | (None, 126, 64) | (None, 124, 128)
Conv1D | output:
dropout_3 | input: | (None, 124, 128) | (None, 124, 128)
Dropout | output:
max_pooling1d | input: | (None, 124, 128) | (None, 62, 128)
MaxPooling1D | output:
flatten | input: | (None, 62, 128) | (None, 7936)
Flatten | output:
dense_2 | input: | (None, 7936) | (None, 50)
Dense | output:
dense_3 | input: | (None, 50) | (None, 6)
Dense | output:

Fig. 11. 1D-CNN network architecture.


Model Accuracy

Model Accuracy

Model Accuracy

Model Accuracy

Fig. 12. Accuracy plots of TOP (left):LSTM model TOP(right): Multi-layer LSTM model, BOTTOM(left): CNN model and BOTTOM(right): 1D-CNN + LSTM model. The blue line shows test data, and the red one indicates training data.

References

[1] Wang, J., Chen, Y., Hao, S., Peng, X. & Hu, L. Deep learning for sensor-based activity recognition: A survey. *Pattern Recognition Letters* **119**, 3–11 (2019). URL https://doi.org/10.1016%2Fj.patrec.2018.02.010.

[2] Vepakomma, P., De, D., Das, S. K. & Bhansali, S. 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)*, 1–6 (2015).

[3] Qin, J., Liu, L., Zhang, Z., Wang, Y. & Shao, L. Compressive sequential learning for action similarity labeling. *IEEE Transactions on Image Processing* **25**, 756–769 (2016).

[4] Hammerla, N. Y., Halloran, S. & Ploetz, T. Deep, convolutional, and recurrent models for human activity recognition using wearables (2016). URL https://arxiv.org/abs/1604.08880.

[5] Moya Rueda, F., Grzeszick, R., Fink, G. A., Feldhorst, S. & Ten Hompel, M. Convolutional Neural Networks for Human Activity Recognition Using Body-Worn Sensors. *Informatics* **5** (2018). URL https://www.mdpi.com/2227-9709/5/2/26.

[6] Zheng, Y., Liu, Q., Chen, E., Ge, Y. & Zhao, J. L. Time series classification using multi-channels deep convolutional neural networks. In *WAIM* (2014).

[7] Ordóñez, F. J. & Roggen, D. Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition. *Sensors* **16** (2016). URL https://www.mdpi.com/1424-8220/16/1/115.

[8] Mario, M.-O. Human activity recognition based on single sensor square hv acceleration images and convolutional neural networks. *IEEE Sensors Journal* **19**, 1487–1498 (2019).

[9] Lee, S.-M., Yoon, S. M. & Cho, H. Human activity recognition from accelerometer data using convolutional neural network. In *2017 IEEE International Conference on Big Data and Smart Computing (BigComp)*, 131–134 (2017).

[10] Liu, K.-C. *et al.* Deep learning based signal enhancement of low-resolution accelerometer for fall detection systems (2020). URL https://arxiv.org/abs/2012.03426.

[11] Hochreiter, S. & Schmidhuber, J. Long Short-Term Memory. *Neural Computation* **9**, 1735–1780 (1997). URL https://doi.org/10.1162/neco.1997.9.8.1735. https://direct.mit.edu/neco/article-pdf/9/8/1735/813796/neco.1997.9.8.1735.pdf.

[12] Das Antar, A., Ahmed, M. & Ahad, M. A. R. Challenges in sensor-based human activity recognition and a comparative analysis of benchmark datasets: A review. 134–139 (2019). URL 10.1109/ICIEV.2019.8858508.

[13] Machado, I. P., Luísa Gomes, A., Gamboa, H., Paixão, V. & Costa, R. M. Human activity data discovery from triaxial accelerometer sensor: Non-supervised learning sensitivity to feature extraction parametrization. *Information Processing and Management* **51**, 204–214 (2015).

[14] Ronao, C. A. & Cho, S.-B. Human activity recognition with smartphone sensors using deep learning neural networks. *Expert Systems with Applications* **59**, 235–244 (2016). URL https://www.sciencedirect.com/science/article/pii/S0957417416302056.

[15] Zhao, Y., Yang, R., Chevalier, G., Xu, X. & Zhang, Z. Deep Residual Bidir-LSTM for Human Activity Recognition Using Wearable Sensors. *Mathematical Problems in Engineering* **2018** (2018). 1708.08989.

[16] Wang, H. *et al.* Wearable Sensor-Based Human Activity Recognition Using Hybrid Deep Learning Techniques. *Security and Communication Networks* **2020** (2020).

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

[18] Zaki, Z., Shah, M. A., Wakil, K. & Sher, F. Logistic regression based human activities recognition. *JOURNAL OF MECHANICS OF CONTINUA AND MATHEMATICAL SCIENCES* **15**, 228–246 (2020).

[19] Mutegeki, R. & Han, D. S. A cnn-lstm approach to human activity recognition. In *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, 362–366 (2020).