# Human Physical Activity Recognition

Mehedi Galib (UG34492)

***Abstract*—Human activity identification has been a significant study issue in the last several years, leveraging machine learning techniques. Using the PAMAP2 data set, the deep neural network developed in this study aims to classify human activities. To identify human actions such as sitting, walking, jogging, climbing and descending stairs, etc., the model framework is built using LSTM networks. The primary objective is to optimize the model's hyperparameters for standard classification accuracy and lowered computation metrics for energy efficiency. In order to provide a comprehensive analysis, the work is compared to the baseline model SensorNet. Moreover, the trained model tested with physical activities recorded by smartphone, known as a smartphone dataset, to justify the efficacy of transfer learning.**

***Index Terms*—LSTM, PAMAP2, DNN, human activity detection.**

## I. Introduction

In recent years, human physical activity recognition are experiencing tremendous growth in a wide range of time series applications, such as health analysis, geriatric health, information security, and human-machine interaction, among others [1]. Among these wide applications, effective classification and modelling of time series data to measure vital indicators (heart rate, blood pressure), fitness and wellness to track activity. These time-series data are typically generated from different sensor modalities, where multimodal signals have different sampling frequencies due to the data measurement procedures used with accelerometers, magnetometers, gyroscopes and heart rate monitors. Hence, accurate analysis and recording of human physical activity can be utilized to evaluate human health and guide human colloquial activity. A smartphone and a wearable gadget or body sensor network typically record daily physical activity. However, there are numerous situations where individuals cannot or are unwilling to use any device. In this instance, cameras and computer vision can monitor physical activity.

Now, in the case of analyzing these data, several machine-learning and deep-learning methods have been proposed for detecting the appropriate features among these collected data and classifying them using traditional or customized classifiers. Traditionally, K nearest neighbors (KNNs) and deep neural networks (DNNs) have been used to address these above-mentioned multimodal time series issues. In particular, Dynamic Time Warping (DTW) [1], K nearest neighbors (KNNs) [2], Deep Neural Networks (DNNs), and End to End Convolutional Neural Networks (End to End CNN). Since classification tasks employing KNNs and DTW are associated with long training times, DNNs have become very popular for multimodal signal processing [3] - [4]. Nevertheless, they, too, require energy-consuming architectures that are not feasible for IoT (Internet of Things) applications. Consequently, DNNs incorporating convolutional neural networks (CNNs) work very well for feature extraction and multi-classification problems. They are extensively used for computer vision applications in classification and recognition. The ability of CNNs to learn features from raw data signals makes them perfectly suitable for these tasks. Despite these valuable traits, one major disadvantage of CNNs is their high memory and computation requirements. To address these drawbacks with CNN, recurrent neural network (RNN) has been proposed in the literature, where recursive feature occurs in the time frame address efficiently.

Hence, several deep neural network models for feature extraction, activity detection, classification, and segmentation are based on the PAMAP2 [2] dataset have been proposed. SensorNet [3] employs CDNN (Convolutional Deep Neural Network) to do multiclass classification, which is the work most relevant to our objectives with the PAMAP2 data set. SensorNet implemented a similar model that takes input from 40 sensor modalities to classify 13 tasks using CNNs [5]. However, as aforementioned, CNNs perform well for feature extraction and multi-classification rather than time series data, where RNN shows exciting improvement with allowable network topology. Hence, we propose an LSTM model for multiclass classification on the PAMAP2 dataset in this project and compare the results to DNN models for thorough justification. Using this architecture as a benchmark, we utilize SensorNet architecture as a baseline and compare our proposed model's performance with SensorNet's physical activity categorization findings. Moreover, we adopt the concept of transfer learning to use the proposed model of Human Activity Recognition Using a Smartphones Data Set. In particular, the key contributions in this work are:

- Proposes an LSTM model for activity classification.
- Perform hyperparameter optimization to reduce power and memory requirements while maintaining standard accuracy.
- Evaluate the model performance in terms of detection accuracy, memory and computation for the case study of Physical Activity Monitoring dataset PAMAP2.
- Employ the concept of transfer learning to use proposed model to Human Activity Recognition Using Smartphones Data Set.

In Section II, the background of proposed work is provided. In Section III, we discuss about our methodology for implementing LSTM model for our proposed system. Section IV describes the results obtained in this work, while Section V concludes this work.

Email:mgalib1@umbc.edu. Special Thanks to Dr. Tim Oates for fruitful discussion related to the project

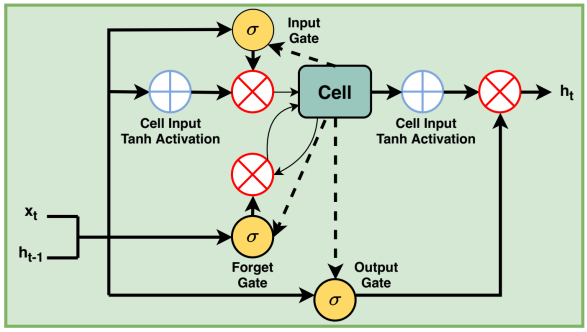


Fig. 1. An LSTM Cell which has input, output, and forget gates. Here $x_t$ is the input, $h_{t-1}$ is the output of the previous state and $h_t$ is the output of the current state [4].

## II. Background & Related work

As aforementioned, numerous research and computational models address real-time-series signals. Complex time series data with many modalities have been used for the careful implementation of monitoring and diagnostic scenarios in medical contexts [7], [8]. CNN-based models have been widely employed for classification and detection tasks in human activity monitoring. However, CNN only extract spatial information from these time-series data. Nonetheless, signals related to physical activity have concentrated periods of fluctuation or pattern throughout the activity, and the windowing procedure for the CNN design considers these sequences during training. On the contrary, the sequential structure of physiological data makes recurrent neural networks (RNNs) extremely effective. RNNs can collect temporal information associated with multimodal data to generate generalized models. In particular, long short term memory (LSTM) networks are used for human activity identification. In [10], LSTMs were utilized to concatenate positive time direction (ahead state) and negative time direction (reverse state) while also permitting residual connections between stacked cells to address the vanishing gradient problem. This increased the recognition rate in both the temporal and spatial dimensions. The ability of LSTMs to avoid lengthy dependencies offers them an advantage over simple RNN architectures for any classification problem; hence, an LSTM-based model architecture is presented in this project.

LSTM addresses the vanishing gradient problem of RNN through adaptation of the gated module. When either the tanh or the ReLU activation function is used, traditional RNNs cannot parse lengthy sequences. RNN networks have developed to have two distinct variants, which are known as GRU (Gated Recurrent Unit) and LSTM, in order to circumvent this problem. The number of gates is the primary difference between these two networks. In case of LSTM, the module has three gates: an input gate, an output gate, and a forget gate. GRUs only has two gates, which are the reset and update gates. Because of this structural distinction, GRU networks may have fewer parameters, computations, and memories than LSTM network.

On the other hand, LSTM networks with a more significant number of gates have a greater degree of control over their correctness. In addition, the vanishing gradient issue that plagues the fundamental RNN design is circumvented by both of these RNN network versions. LSTM unit architecture is shown in Fig. 1. As mentioned before, the LSTM unit has an input gate, and a forget gate. The input gate decides which memory content should be added, and the forget gate works out the initial cell state necessary for future inference. Although both gates operate in conjunction with one another, the information connection between them can in no way be said to rely on the other. In addition, LSTM networks often use the sigmoid and tanh activation functions.

## III. Methodology

There are a total of 52 features present in PAMAP2 [5], which contains raw sensory data for 13 distinct activities. However, IMU orientation measurements and timestamps will be removed from the dataset due to the likelihood of overfitting. As we know, reducing features reduces computing time and parameters, hence, by restricting the amount of features being used in our LSTM model, we try to decrease both the computation time and the number of parameters needed to be trained. In order to predict multi-class classification of the given data, the proposed model uses a long short-term memory (LSTM) layer to learn the patterns in the sensory input. In order to find an appropriate timestep for the LSTM model, a windowing technique will be used during the dataset's preparation phase.

### A. Dataset

*1) Physical Activity Monitoring Dataset (PAMAP2):* PAMAP2 [5] from UCI Machine Learning is the primary dataset. This dataset incorporates data from 9 subjects performing 13 activities, including lying, sitting, standing, walking, computer work, ascending and descending stairs, etc. Each subject followed a protocol with 12 activities and one label representing the transitory interval between activities. Three Colibri wireless IMUs (inertial measuring units) were worn on the chest, on the dominant side's ankle, and over the dominant arm's wrist for 100 Hz sampling. A 9-Hz cardiac monitor was also used. Data files include one time-stamped and labeled sensory data line. The data files have 54 columns, each with a timestamp, an activity label (ground truth), and 52 sensory data properties. The subjects' raw sensory data has been combined into one dataset. Then the dataset is randomly partitioned into three sets: 80% for training, 10% for validation, and 10% for testing.

*2) Smartphones Dataset:* Thirty 19-48-year-old participants participated in the trials [6]. Each person conducted six activities (walking, walking upstairs, walking downstairs, sitting, standing, laying) while wearing a Samsung Galaxy S II on their waist. At 50Hz, the accelerometer and gyroscope measured 3-axial linear acceleration and 3-axial angular velocity. Video-recorded experiments labeled data manually. After applying noise filters, accelerometer and gyroscope data were sampled in fixed-width sliding windows of 2.56 sec with 50% overlap (128 readings/window). A Butterworth low-pass filter divided the sensor acceleration signal into gravity and body

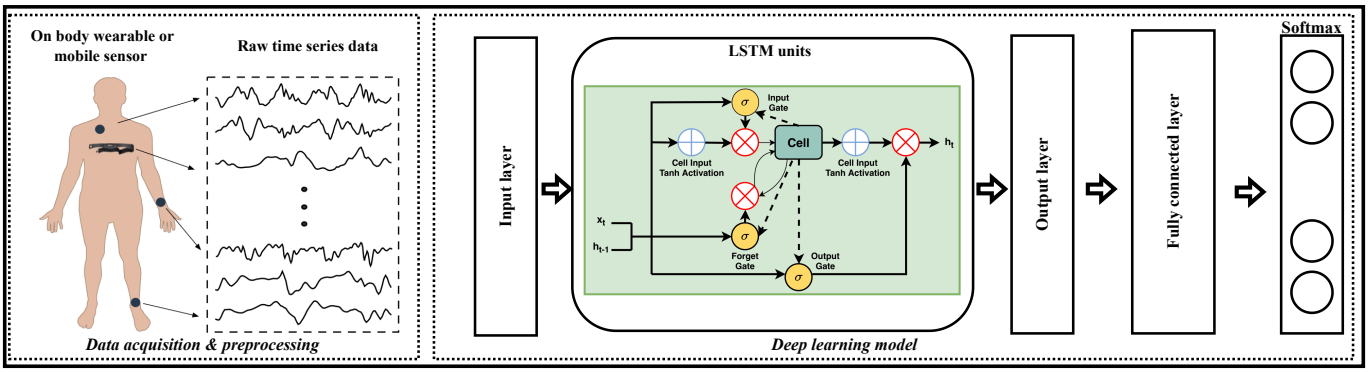


Fig. 2. High-level diagram of the proposed model, where signal are collected from wearable/mobile sensor; then passed to the pre-proceessing unit, where time series data generates, and LSTM followed by the fully connected and softmax layers to classify our desired classes.

acceleration. A 0.3 Hz cutoff filter was selected since the gravitational force is considered to have only low-frequency components. Calculating time and frequency domain variables gave each window a vector of characteristics.

### B. System Model

Our model architecture takes raw time-series signals as input. Nevertheless, as part of pre-processing, the raw signals are processed as window images before feeding them to the model. The architecture determines correlations between sensor modalities using these window images or snapshots of the raw data. Fig. 2 shows a high-level block diagram of the proposed system illustrating the pre-processing, deep neural network module, and softmax layers.

*1) Pre-processing of Sensors Data:* Raw time-series signals consist of $F$ features with the same or different sampling frequency. To generate an image from the variables, a sliding window of size $W$ and increment-step $I$ is passed through all variables, creating a set of images of shape 1. The label associated with this image depends on the dataset. Since a single label is assigned to each image, the label of the current time step is taken as the label of the image . A given image generated at time-step t has the prior states of each variable from $(tW+1)\ldots t$. Thus, the network can look back W prior states of each variable and given the current state of each variable, predicts the label.

*2) Deep Neural Network Model:* Fig. 3 shows the deep neural network architecture implemented in this work. It contains one LSTM block followed by three fully connected layer blocks. The last fully connected layer block is a softmax layer with a size equal to the number of class labels. The window images created during the pre-processing are fed to the LSTM layer for feature extraction. These extracted features are then fed to the fully connected and softmax layers. The LSTM layer consists of 256 neurons and has a timestep of 64 for this case study. The other part of the input to the LSTM block is the feature which is the number of multimodal channels or signals in raw time-series signals. For this physical activity case study, the number of relevant features is 40. Three fully connected layers are utilized in this model architecture, with the first and second ones containing 80 and 32 neurons, respectively. The last layer has a size equivalent to the class labels with a softmax activation. Tanh activation is used in the LSTM layer with a recurrent activation of the hard sigmoid. In all the fully connected layers, ReLu activation is preferred except the last one. The network is optimized using the optimizer Adam. Also, categorical cross-entropy is used as the loss function.

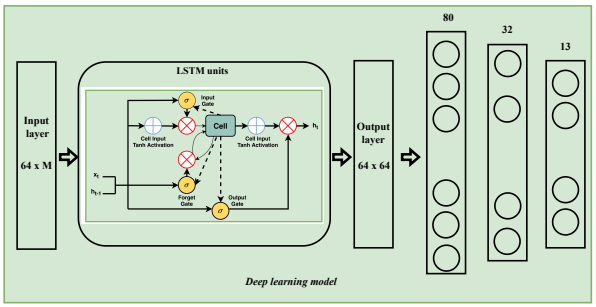


Fig. 3. The proposed network architecture consisting of different layers. The network takes a 1-channel image as input with 1 LSTM and 3 fully connected layers.

*3) Transfer-learning model:* As aforementioned, we train our DNN model with the PAMAP2 dataset and then apply the transfer learning model for the smartphone dataset. In the case of the PAMAP2 dataset, we have 13 different activities, and hence, we employ 13 neurons in the softmax layer. However, since the smartphone dat set contains only six distinct classes, we replaced the feedforward layer of our proposed DNN model with 32 and 6 layers and then inferred with the smartphone dataset.

## IV. Result & Discussion

This section evaluates the model using a real-world case study, including the physical activity monitoring dataset [6] and the smartphone dataset. Moreover, in-depth analysis and experimental results are provided. The model architecture is trained using the py-touch framework. Two different configurations were analyzed: intra-classification, where we split the dataset into the train, dev, and test domains, with a ratio of 80%, 10%, and 10%, respectively. The other configuration involved leaving one subject out for a test at each time, which was done using K-fold cross-validation. However, in the smartphone dataset, we only consider the evaluation dataset to find the efficacy of the transfer learning.

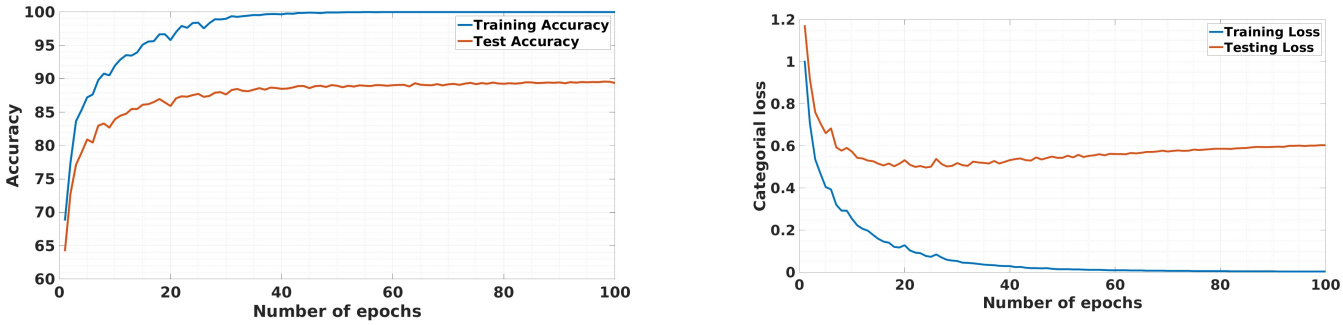


Fig. 4. Change in accuracy and loss with number of epoch. (a) Accuracy vs number of epoch; here training accuracy reaches 100 after 40 epochs, however, testing accuracy struck around 90% (b) Categorical loss vs number of epochs; where again testing accuracy defer with number of iterations.

### A. Experimental setup and data set

As aforementioned, for the experiment, the dataset for each subject was split to have 80% training data, 10% testing, and 10% validation data for PAMAP2 dataset. First, we discuss hyperparameter optimization, since optimized hyperparameters ensure adequate memory allocations while achieving high classification accuracy. The impact of changing the following parameters have been explored here: 1) number of epochs, 2) choice of timestep, 3) intra classification results, and 4) leave one subject out (LOSO) classification results.

*1) Number of Epochs:* The initial stage in optimizing involves determining the number of epochs necessary for training the model. In this instance, we used a preliminary model devoid of optimized parameters to determine the trend for loss and precision at various epochs. The model was trained for 100 iterations to determine the required epoch value while monitoring the validation results for accuracy and loss. Fig. 4 indicates that accuracy stabilizes after 100 epochs. We thus chose to tweak all of our settings over 100 epochs.

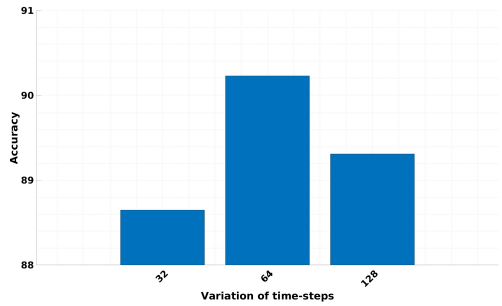


Fig. 5. The overall accuracy results with different timesteps.

*2) Variation of Timestep of LSTM:* The timestep in deep neural networks using LSTM modules determines the temporal information. The timestep is the number of sequences the LSTM block considers while predicting the future. Consequently, the timestep is a crucial parameter that dictates the structure of the whole model. For human activity identification challenges, the timestep should be designed so that each action's sequence of time-series information is accessible within the given timestep duration. As stated in the preceding sections, the sampling frequency of the sensors in this dataset was 100 Hz, which equates to 100 data samples per second. Consequently, if a timestep size of 128 is used, sequences of information for related classes will overlap, which is undesirable. If a timestep size of 32 is chosen, the sequence will be too tiny for the LSTM model to generate an accurate enough prediction. Consequently, a timestep size of 64 is selected for this case study, which gives the highest level of precision given the limitations, as shown in Fig. 5.

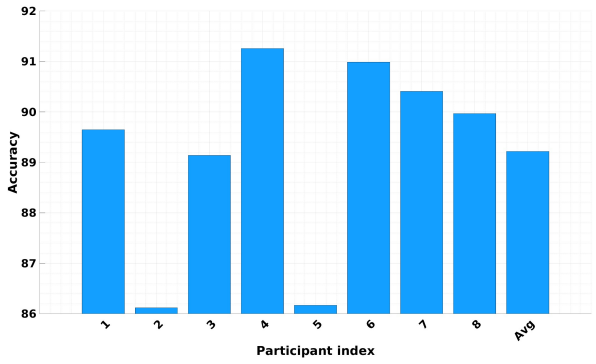


Fig. 6. Results for intra class classification accuracy for timestep of 64 with 100 epochs

*3) Intra Classification Results:* The results of intra-categorization by topic are shown in figure 6. The accuracy represented by the bar graph is 89.22Some topics are less accurate than others (subjects 2 and 5): The fundamental explanation is that these two participants only participate in some activities.

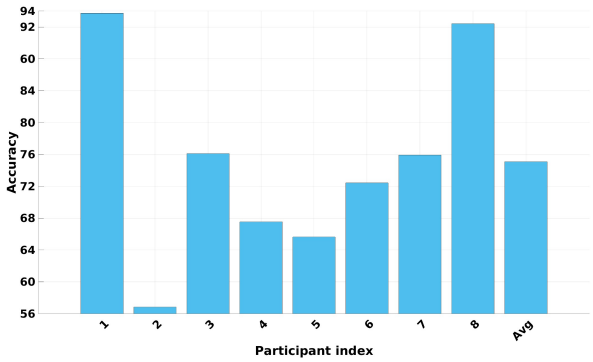


Fig. 7. Results for LOSO Classification with a timestep of 64.

*4) Leave One Subject Out (LOSO) Classification Results:* In this experiment, k-fold cross-validation was used, and a random subject was removed from the training set during each fold. This topic was then evaluated. Therefore, there are eight distinct configurations in which each subject serves as the test set. Such a distinct configuration was performed to guarantee no data leaks between the train and test data. n Fig. 7, we demonstrate the finding of LOSO, where the X-axis represents the subject used as the dataset. The average accuracy score shown by the findings is 75.08%. The lower score results from the fact that there is only eight participants' worth of data, which needs to be improved to develop a generalized model. As the data distribution from each patient changes significantly, testing on brand-new datasets results in a significant decrease in precision. However, the model still achieves an accuracy of 75.08%, indicating that it is still learning, which is better than chance and could be generalized if there were additional subject-specific data.

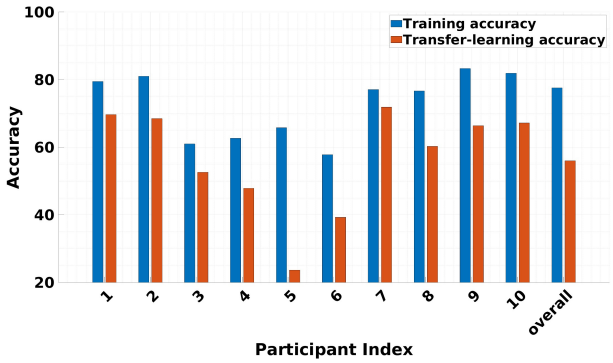


Fig. 8. Transfer learning using Smartphone data-set. Only first ten participants accuracy are listed for the sake of brevity.

*5) Testing on Smartphone Dataset:* In Fig. 8, we have demonstrate the efficacy of transfer learning using smart-phone data-set. For the sake of brevity, we only show the results for first ten participant out of thirty. It is to be noted that, the accuracy of transfer learning is very less compared to training newly with only 20 epochs.

## *B. Model Optimization*

In this section, the model's hyperparameters have been extensively optimized in order to reduce memory needs while attaining excellent classification accuracy. The effects of modifying the following factors are investigated here: 1) number of LSTM layer neurons, 2) batch Size, and 3) activation functions.

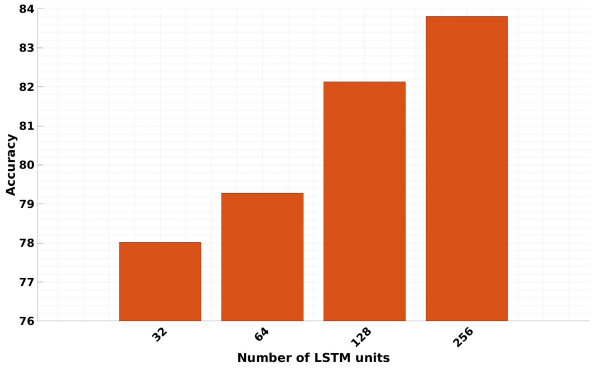


Fig. 9. Variation of Accuracy with Different Neurons for the LSTM Block.

*1) Number of neurons in the LSTM layer:* The number of neurons for the LSTM block has been set to 256 in this experiment. This value was determined by iteration. The model was trained with 32, 64, 128, and 256 LSTM neurons with a batch size of 1024. Figure 9 shows the performance of the model using several LSTM neurons. It is evident that, even though low neurons may have fewer parameters, accuracy requires a big suggestion. In addition, the accuracy findings pertain to intra-classification, and the score would be significantly lower if LOSO were added. Therefore, 256 neurons offer the optimal balance between accuracy and memory restrictions.

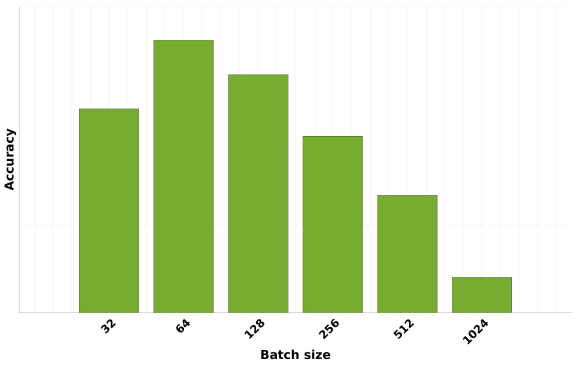


Fig. 10. Impact of increasing the Batch Size for the model.

*2) Batch Size:* Batch size represents the number of data the model considers prior to updating the weights. The larger the batch size, the greater the number of samples and the faster the calculation. The weights are kept the same as in a model with a smaller batch size, resulting in a decrease in accuracy. According to figure 10, 64 was chosen as the batch size for this case study because it delivers the highest accuracy score among all batch sizes. Also, if the batch size is too small, the

TABLE I
COMPARISON OF MODEL PERFORMANCE WITH RELATED WORK

| Metrics/Technique | SensorNet [3] | This Work |
|---|---|---|
| **Accuracy** | **86** | **89.22** |
| **Number of Parameters** | **175K** | **2.9M** |
| **Number of Operations** | **50M** | **3.4M** |
| **Memory Requirements** | **219.6 KB** | **5.12 MB** |

accuracy of an LSTM model decreases. This was also shown with the batch size of 32.

*3) Activation Function:* Controlling the LSTM cell are an input gate, and a forget gate layer. When LSTM looks back to anticipate the future, it chooses which information to retain and which to discard. Traditionally, the forget gate layer is a sigmoid layer that generates a number between 0 and 1 for each input. This layer is responsible for forgetting information. A 1 indicates "totally retain this," whereas a 0 indicates "absolutely discard this." Using the tanh layer, the newly added candidates to the cell state are generated. The activation and recurrent activation for the LSTM block were set to tanh and hard sigmoid, respectively, for this experiment. If nonlinear activation functions such as tanh and sigmoid are used, layers in deep neural networks often backpropagate mistakes to update the weights. This is known as the issue of disappearing gradients. For the fully linked layer, the activation was thus adjusted to Rectified Linear Unit (ReLu) to prevent the gradient-vanishing problem. ReLu is linear for positive values, and returns zero for negative values.

*4) Bench-marking with existing Methods:* In this section, we compare the performance of the suggested model to the related work SensorNet. SensorNet applies a deep convolutional neural network to the PAMAP2 dataset for categorization. Table I demonstrates that CNNs have a small number of parameters but do significant work. However, because of the smaller number of parameters, CNNs have a substantially smaller memory footprint than LSTM networks, which need a more significant amount of memory to forecast the future by recalling past events. The suggested LSTM-based model achieves 89.22% accuracy, more than SensorNet for the 64-step window and step size.

## V. CONCLUSION

Using LSTM networks, the project proposes a categorization strategy for physical activity. First, the raw time-series signals are transformed into window pictures for use with the LSTM module. The sensor modalities are then classified using the LSTM-based DCNN. The suggested architecture achieves 89 % intra-classification accuracy and 75 % LOSO classification accuracy. The task also includes the optimization of the modules' parameters and the justification of the parameter-setting procedure. Even though the suggested model takes more memory for categorization than the previous approach, the scheme's primary result is an improvement in accuracy.

## REFERENCES

[1] L. Mo, F. Li, Y. Zhu, and A. Huang, "Human physical activity recognition based on computer vision with deep learning model," in *2016 IEEE international instrumentation and measurement technology conference proceedings*. IEEE, 2016, pp. 1–6.

[2] A. Reiss and D. Stricker, "Creating and benchmarking a new dataset for physical activity monitoring," in *Proceedings of the 5th International Conference on PErvasive Technologies Related to Assistive Environments*, 2012, pp. 1–8.

[3] A. Jafari, A. Ganesan, C. S. K. Thalisetty, V. Sivasubramanian, T. Oates, and T. Mohsenin, "Sensornet: A scalable and low-power deep convolutional neural network for multimodal data classification," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 1, pp. 274–287, 2018.

[4] A. N. Mazumder, H.-A. Rashid, and T. Mohsenin, "An energy-efficient low power lstm processor for human activity monitoring," in *2020 IEEE 33rd International System-on-Chip Conference (SOCC)*. IEEE, 2020, pp. 54–59.

[5] A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," in *2012 16th international symposium on wearable computers*. IEEE, 2012, pp. 108–109.

[6] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine," in *International workshop on ambient assisted living*. Springer, 2012, pp. 216–223.

## VI. MISCELLANEOUS

### A. Data Source

- The PAMAP2 dataset can be found here: URL: https://archive.ics.uci.edu/ml/datasets/pamap2+physical+activity+monitoring
- Human Activity Recognition Using Smartphones Data Set URL: https://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones