Comparative Analysis of Deep Learning Architectures for Human Activity Recognition: A Study of Conformer-based Networks vs CNN and BiLSTM Networks

Vijay Kumar - 40221480
COEN 691 – Pervasive Computing for Healthcare
Department of Electrical and Computer Engineering
Concordia University
Montreal, Canada
Vijay-_kumar@hotmail.com

I. INTRODUCTION

The project focuses on Human Activity Recognition (HAR) using Inertial Measurement Units (IMUs). IMUs offer precise motion and orientation measurements. There are a wide range of applications revolving around HAR such as applications in healthcare, surveillance, sports and fitness tracking. Sensor based recognition typically uses built in sensors in embedded devices, the advantage with such type of sensors is that it is low power consumption, lightweight, and most importantly IMU sensors embedded can capture movements related to the action performed reliably, efficiently and accurately.

The goal is to develop an efficient model capable of recognizing various activities using data collected from accelerometers and gyroscopes. The methodology used in this scientific exploration revolves around analyzing deep learning methods and comparing their performance in activity recognition. Deep learning models in particular perform exceptionally well when dealing with non-linear problems, but the performance is heavily reliant on the quality of data and model architecture. Datasets are typically imbalanced and contain noise and if used with a machine learning model, it can lead to a model which either has a bias towards the class with highest number of samples or would misclassify a certain pattern due to the noise induced. The architecture proposed specifically handles such problem and is designed to overcome such obstacle and achieve accurate recognition performance regardless of the imbalance or noise in data.

The learning goal of this comparative analysis is to gain hands on experience in implementing deep learning models, preprocessing data, and building pipelines for activity recognition. Overall the project aims to contribute insights into building HAR systems and understanding of deep learning methodology for HAR applications.

A. Selected papers description

The paper titled “An optimized deep learning model for human activity recognition using inertial measurement units” [1] proposes a solution that combines Convolutional Neural Networks (CNNs) and Bidirectional Long short term memory networks (Bi-LSTMs) into a single architecture. The network architecture leverages the CNN to extract spatial features from the raw sensor data, and uses the Bi-LSTMs to capture temporal patterns. Using both types of features, allows the model to effectively recognize complex activity patterns from raw sensor data regardless of the noise introduced or imbalances in data. The second paper titled “Conformer-Based Human Activity Recognition using Inertial Measurement Units ” [2] proposes an architecture which uses CNNs, multiple conformer blocks, residual connections, and a self-attention module within a transformer network. This architecture proposed minimizes the number of parameters in the network, and each of the mentioned blocks in the proposed architecture serves a purpose. The attention mechanisms within conformer blocks effectively capture temporal features, and residual connections prevent gradient vanishing issues, facilitating model convergence.

B. Dataset description

Two datasets are selected to evaluate the proposed model, the MHealth dataset consists of data collected via multiple sensors attached at different locations of the human body along with vital sign recordings connected to the chest sensor that uses 2-Lead ECG measurements. In total 23 features correspond to sensor data collected from sensors attached to the Chest, Left Ankle and Right arm and 2 measurements which corresponds to the vital health readings by the ECG sensor. The dataset contains recordings from 12 activities performed which were recorded at a sampling rate of 50Hz. Each row corresponds to a single measurement at a timestamp, the data type used to store such readings are float values.

	user	a_rarm_x	a_rarm_y	a_rarm_z	g_rarm_x	g_rarm_y	g_rarm_z	label
0	0	-6.7368	-6.6124	2.7631	-1.2333	-0.47844	-0.051724	0
1	0	-6.4254	-6.5098	3.0183	-1.2333	-0.47844	-0.051724	0
2	0	-6.5062	-6.7122	2.779	-1.2333	-0.47844	-0.051724	0
3	0	-5.906	-6.7621	2.7869	-1.249	-0.46817	-0.045259	0
4	0	-6.7218	-6.8831	2.6871	-1.249	-0.46817	-0.045259	0

Figure 1.MHealth Dataset sample

The WISDM dataset [3] consists of raw accelerometer and gyroscope data collected from smartphones and smartwatches at a sampling rate of 20Hz. 51 subjects performed 18 activities for 3 minutes each. Each recording holds 6 features corresponding to accelerometer and gyroscope measurements in all 3 axes. Each row from the dataset represents a single measurement at a timestamp which holds accelerometer and gyroscope measurement in all 3 axes along with the actual label of the activity.

	user	activity	timestamp	acc_x	acc_y	acc_z
0	33	Jogging	49105962326000	-0.6946377	12.680544	0.50395286
1	33	Jogging	49106062271000	5.012288	11.264028	0.95342433
2	33	Jogging	49106112167000	4.903325	10.882658	-0.08172209
3	33	Jogging	49106222305000	-0.61291564	18.496431	3.0237172
4	33	Jogging	49106332290000	-1.1849703	12.108489	7.205164

Figure 2. WISDM Dataset sample

II. METHODOLOGY

A. Preprocessing Steps

Preprocessing the required data is done in a 4 step process whose outcome is a clean set of data that can be used by the machine learning model. The stages required are as such:

1. **Filtering:** In order to remove noise a filter should be applied; the low pass filter is combined with a median filter to remove noise. The median filter works by replacing each data point with the median value within its selected neighboring window size, by doing so the signal is smoothed and edges/features are preserved. In this two-stage process, median filter is applied first and then the low pass filter with a cutoff frequency of 2Hz is applied, results can be observed in the figure shown below.

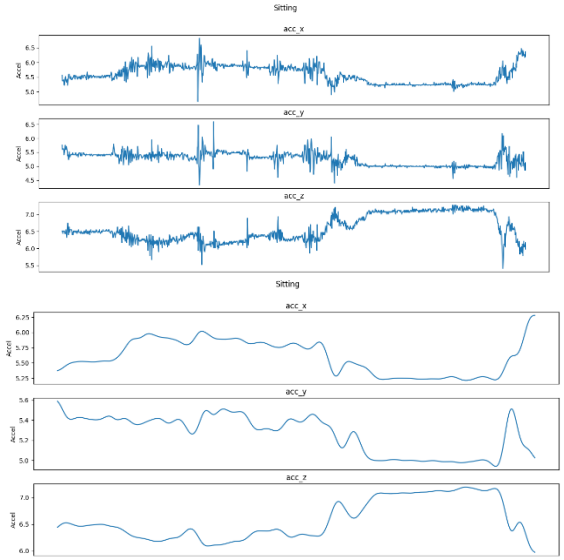

Figure 3. Noisy signal (Top) - Filtered Signal (Bottom)

2. **Segmentation:** A sliding window of size 200 is applied to extract segments of sensor data with an overlap of 50% This allows capturing of trends or patterns that might be missed in a continuous data stream. It also enhances the model's ability to extract useful features and generate feature maps for machine learning.
3. **Standardization:** Z-score normalization transforms the dataset to have a zero mean and a standard deviation of 1. this ensures comparability between sensor readings and aids in better machine learning model convergence. The Z-score equation can be defined as such:

$$z = \frac{x - \mu}{\sigma} \quad (1)$$

Where z is the z-score or standardized value, x is the data point, μ is the mean of the dataset, and σ is the standard deviation of the dataset.

4. **Splitting dataset:** To train the machine learning model, the dataset is split into 3 sets – Training set, Validation Set, and Testing set. For the selected application, a 70-20-10 ratio was used for the MHealth dataset and as for the WISDM dataset 80-10-10 split was used. To avoid data leakage and enhance generalization of model, the proposed approach is to split the data based on the number of subjects recorded, meaning that if 10 subjects were present and each of the subjects have performed 7 different activities each, then 7 subjects along with their recorded data will be used as training, 2 subjects will be used as validation and the remaining subject will be used as testing.

B. Model Architecture Implementation

A detailed explanation on each of the selected architectures will be provided below, highlighting the important blocks and modules required to implement and deploy such model.

1. Conformer based DNN with self-attention mechanism

The conformer-based model [2] [4], uses a series of modules that when combined forms the conformer block which is a convolution augmented transformer block that was developed by google. Figure 4 below is a representation of the architecture proposed. The implementation of such model requires attention layers, convolutional layers, linear layers, activation functions, and normalization layers. The conformer block utilizes residual (Skip) connections to concatenate data from the previous layer onto the output of the current layer. This primarily deals with the issue of vanishing gradients.

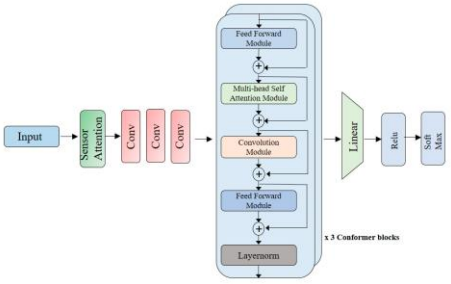

Figure 4. Conformer DNN Architecture

The conformer block can be implemented in blocks as each block shown in Figure 4 above represents a series of layers connected to each other. The Feed Forward Module (FFN) [4] is a block responsible for transformation of data using normalization and swish function. The swish function can be defined as such:

$$f(x) = x * \sigma(\beta x) \quad (2)$$

Where σ represents a sigmoid function, β is a trainable parameter.

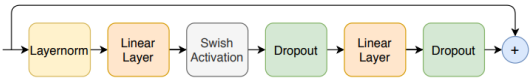

Figure 5. FFN Module

The second type of block used in this architecture is the Multi-head Self attention module (MHSA) which is responsible to allow simultaneous processing of different input sequences by employing the use of multiple attention heads. a diverse representation and relationships can be built as patterns captured reveal a much better correlation between the types of data collected. **Figure 6** shown below is an implementation of what layers are used to build this module.

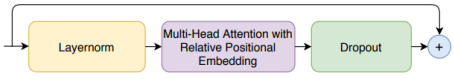

Figure 6. MHSA module

Finally, the Convolutional module (CONV) which is responsible for the features extracted can be implemented as shown in **figure 7**. The main layers in this block are the convolutional layers and activation functions used to transform the data into useful representations. A series of convolution and transformations are applied to extract and generate a collection of feature maps that can be used for classification purposes.

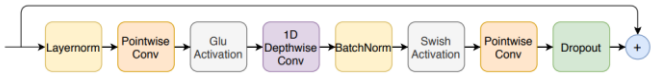

Figure 7.CONV module

By combining all the defined modules, the conformer block can be built and as a result the whole architecture becomes a series of modules put after each other which in mathematical terms can be represented as such:

$$\begin{aligned} \tilde{x_i} &= x_i + \frac{1}{2} FFN(x_i) \\ x_i' &= \tilde{x_i} + MHSA(\tilde{x_i}) \\ x_i'' &= x_i' + CONV(x_i') \\ y_i &= Layernorm\left(x_i'' + \frac{1}{2} FFN(x_i'')\right) \end{aligned} \quad (3)$$

Table 1 are the hyperparameter values selected to train the model

Table 1. Conformer model hyperparameters

Variable	Value
Convolution filters	256
Kernel size	3
Optimizer	Adam
Learning Rate	0.0001
Epochs	50
Data Split Ratio (train: Val: Test)	80:10:10

2. CNN + Bi-LSTM Neural Network architecture

The proposed architecture merges a convolutional neural network and a Bi-LSTM network into a single model to process and extract features from the sensor data. The CNN uses 1D convolutional to generate feature maps while the Bi-LSTM layers uses analyze the sensor data by propagating the data forward and backwards in both direction. When both networks are combined, the results allow to capture spatial and temporal features from the sensor data which can be used as patterns for the classification layer. For the implementation of the CNN network, only convolutional layers and pooling layers are required. Convolving results in feature maps generated and pooling results in reduction of dimensions of the feature maps extracted. The Bi-LSTM layer similarly uses only a single type of layer which propagates the data in both directions and extracting any useful patterns. To conclude, the implementation of both architectures was done using TensorFlow as the backend and Keras libraries as the required layers and activation functions are already available in the library described.

Table 2. CNN + Bi-LSTM model parameters

Variable	Value
N1	64
N2	64
F1	5
F2	3
B1	128
B2	64
D1	32
Optimizer	Adam
Learning Rate	0.002
Epochs	150
Data Split Ratio (train: Val: Test)	70:20:10

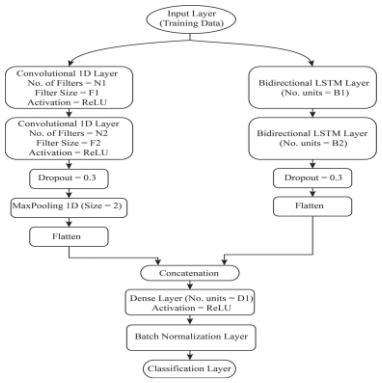

Figure 8. CNN + Bi-LSTM Architecture

III. FINDINGS & RESULTS

A. Main results

1. Evaluation Metrics

Accuracy was primarily used during training to monitor performance, measuring overall performance. In addition, precision, recall, and the F1-Score were used to assess overall performance. Accuracy measures correct predictions over total observations, precision evaluates correct positive predictions, recall measures correct positive predictions over actual positives, and the F1-Score provides a harmonic mean of precision and recall, which is beneficial for unbalanced datasets. The main metric used to assess and track the performance of the model was the accuracy as it measures overall correctness and gives a rough estimate on how the model is performing. Following that, the remaining metrics which are considered as secondary metrics for this application are used to assess the performance of the model given when new unseen data is provided, meaning that by using precision we can observe the model's ability to correctly identify true cases, by using recall we can assess the model's ability to avoid false negatives, and the f1 score is a useful metric when dealing with unbalanced datasets.

2. Conformer based DNN with self-attention mechanism

Evaluating the performance on two sets of unseen data: one consisting of 10% of the training dataset which hold 1 users data collected which is unsee from the model, and the other representing ground truth data collected for evaluation. During data collection, participants performed five activities: walking, upstairs, sitting, standing, and downstairs. Some misclassifications occurred, and the obtained results are detailed below.

	Precision	Recall	F1 Score
0	0.56	0.25	0.34
1	0.83	0.88	0.86
2	1	1	1
3	0.67	0.6	0.63

4	0.37	0.45	0.4
5	0.82	0.87	0.85
Accuracy on test set 1			77 %
Accuracy on test set 2			23%
Proposed Model Performance in paper			98.2 %

3. CNN + Bi-LSTM Neural Network

Similarly, two sets of unseen data are used to evaluate the model, the first set being a user's data with multiple activities kept aside which is unseen by the model, and the second set which is the ground truth used in the above evaluation.

	Precision	**Recall**	**F1 Score**
0	1	1	1
1	0	0	0
2	1	1	1
3	1	0.73	0.85
4	0.36	0.93	0.51
5	0.89	0.24	0.38
6	0.97	1	0.99
7	0.66	0.97	0.78
8	1	1	1
9	0.97	1	0.98
10	1	1	1
11	1	0.91	0.95
Accuracy on test set 1			81 %
Accuracy on test set 2			49%
Proposed Model Performance in paper			99.28 %

B. Data Collection Experience

Accelerometer and Gyroscope data was collected during the collection phase, the Sensorbox mobile application was used and by leveraging the integrated sensors in mobile phones data was collected. Two subjects each performed five activities each, resulting in a total of 10 recorded samples. Quality of data was a primary concern, with activities monitored closely and recorded individually. Each activity was saved as a CSV file and each activity lasted between 12 to 20 seconds. To introduce variability, activities were performed at different paces across trials. The data collection process provided valuable exposure on understanding how controlled data collection is performed, making sure that everything is in alignment with activity protocols and sensor functionality. Privacy was prioritized, and quality control measures were maintained to ensure the collection of high-quality data. Overall, the process of data collection was a systematic approach which allowed a proper way of collecting and acquiring data.

C. Learning outcome & future plan

Gaining insight into how to build an effective pipeline for HAR revealed crucial steps for building a robust HAR system. Preprocessing sensor data is a pivotal stage, making sure that noise is removed and data is formatted according to the required format needed for machine learning models. Various transformations applied during preprocessing significantly impacts the model's performance, potentially enhancing activity recognition accuracy. Exploring how to build deep learning models allowed me to understand how there should be a balance between optimal parameter selection and model performance so that the scenario of overfitting or underfitting can be avoided. Additionally, implementing complex deep learning architectures revealed various approaches for feature extraction. Understanding how to build an effective HAR pipeline provided comprehensive insights into real-world application deployment. Moving forward, the goal is to optimize model performance by experimenting with diverse architectures, reducing model complexity, and incorporate more variability of data in order to capture a diverse set of patterns that could be extracted by the machine learning models that could further lead to a better performing model.

4. COMPARISON OF PROJECTS

STEP STRIVE is an application designed to recognize real-time activities using data from accelerometer and gyroscope sensors. Unlike my project which involves complex deep learning models, STEP STRIVE employs a simpler model which is random forest algorithm trained on the WISDM dataset. This selection of model reduces computational power required while allowing real-time activity prediction. While STEP STRIVE effectively identifies basic activities in real time, its limitation lies in recognizing a limited set of activities. On the other hand, the scientific project "A Similarity-based semi-supervised learning for activity recognition" focuses on evaluating model transferability and performance on unseen data. Unlike my approach, this project utilizes simple models such as random forest, decision trees, KNN, and naïve Bayes. While these models require less computational power, they struggle with feature engineering and robustness against imbalanced and noisy data.

In contrast, my implementation uses complex deep learning architectures capable of handling data imbalances and noise. Although the model may struggle with certain activities, optimizing hyperparameters, reducing model complexity, and employing feature engineering techniques can enhance the performance. As such for future improvements hyperparameters will be optimized and architecture should be further optimized.

IV. REFERENCES

[1] S. K. Challa, "An optimized deep learning model for human activity recognition using inertial measurement units," *Expert Systems Wiley*, 2023.

[2] S. Seenath, "Conformer based human activity recognition using inertial measurement units," *Sensors*, 2023.

[3] G. M.Weis, "WISDM Smartphone and Smartwatch activity and biometrics dataset".

[4] A. Gulati, "Conformer: Convolution-augmented Transformer for speech recognition," *Google Inc.*, 2020.