Multivariate Time-Series Classification of Daily and Sports Activities using Deep Learning

Faezeh Sabzialiabadi

Department of Applied Data Science
Carinthia University of Applied Sciences
Villach, Austria

sabzialiabadi.Faezeh@edu.fh-kaernten.ac.at

***Abstract*___Human Activity Recognition (HAR) from sensors matters for health monitoring, rehabilitation and sports analytics. Human Activity Recognition (HAR), from sensors is useful. We focus on classification of daily activities and sports activities. We work with multivariate time-series data that come from five body-worn sensor units. The five body-worn sensor units include an accelerometer, a gyroscope and a magnetometer. Deep learning models for sequence modeling are evaluated. A split by subject is used to test how the models work on people not seen before. The models included an one-dimensional CNN, a multi-sensor CNN that has parallel branches for each body location, Temporal Convolutional Networks (TCN) and tuned versions of TCN and a hybrid model that combines a CNN with a BiLSTM. This project measured performance, with test accuracy, precision, recall and F1 and it looked at the confusion matrix. Overall, we observed that multi-sensor feature-level fusion got the results. We see that learning location- representations before combining the representations makes the performance more reliable, for this dataset.**

Keywords—Human Activity Recognition, CNN, Wearable Sensors. Time Series

I. INTRODUCTION

Human Activity Recognition (HAR) infers a person's activity from sensor signals. Human Activity Recognition (HAR) helps with health monitoring, rehabilitation, and sports analytics. In real situations, sensor data are multiple sensor streams that include different types of sensors, such as, accelerometer, gyroscope, magnetometer. Human Activity Recognition (HAR) deals with data that is too big to read by eye.

Earlier HAR systems often used hand-made features such as frequency descriptors, and HAR systems used the classical classifiers [1]. Recently, deep learning models have become popular because they can learn useful representations directly from the raw time-series data. Specifically, the 1D convolutional networks performs well for learning time patterns. Recurrent and temporal convolutional models, such as LSTM/TCN, capture time relationships. We find that another common direction is multi-sensor fusion. Multi-sensor fusion combines signals from the body locations to improve robustness.[2]

Even though there has been progress, HAR remains challenging when the goal is generalization to unseen subjects: Different people may do the activity with different strength and style and some activities produce similar motion patterns. I see that HAR becomes even harder in multi-unit setups because the useful signals, in multi-unit wearable setups are spread across body parts.

Therefore, the goal of the project is to classify 19 sports activities from multiple wearable time series that are collected at five body locations. To address the problem, this project evaluates time-series deep learning architectures using a subject-wise split so that the subject-wise split prevents identity leakage. The study begins with a 1D-CNN. Then the study compares temporal models such as, TCN variants and CNN+BiLSTM. The study also tests a multi-sensor fusion CNN that processes each body unit in a separate branch before the branch combines the features. We measure the model performance with accuracy scores across classes and confusion matrices. The model performance measurement lets us see the misclassifications. We look at the confusion matrices to find the mistakes in model performance.

I. RELATED WORK

Human Activity Recognition (HAR) from wearable sensors is typically formulated as a multiclass classification task on multivariate time-series signals. Earlier approaches used the handcrafted statistical and frequency domain features, such as the mean, the variance, and the FFT-based descriptors. Those approaches then used the classifiers, like SVM or Random Forest. But feature engineering takes a lot of time and often misses the temporal structure [3].

More recent studies increasingly use deep learning to learn features directly from raw sensor sequences. The deep learning models include 1D-CNN models that are widely used in HAR because the 1D-CNN models extract time patterns from the multi-channel inputs [4], [5]. Recurrent models such as LSTM/GRU form another group designed to model longer temporal dependencies in sequential data [6]. Hybrid structures that combine CNN and LSTM/GRU have also been proposed. The CNN layers in the structures learn local features and the recurrent layers, in the hybrid structures capture longer range context.[7]

Another research direction is multi-sensor fusion, where signals from different body locations or sensor modalities are combined to improve robustness and recognition accuracy. Fusion can be performed at the input level (early fusion) or at the feature/decision level (late fusion). It depends on the architecture. Recently, Temporal Convolutional Networks (TCNs) have been adopted for sequence modeling using dilated convolutions and residual connections and providing an alternative to recurrent networks. It shows competitive performance [8].

Based on these trends, this project compares baseline 1D-CNN with temporal models (TCN variants and CNN+BiLSTM) and a feature-level multi-sensor fusion architecture. The project tests the generalization of each model under a wise split.

II. DATA DESCRIPTION

In this project the Daily and Sports Activities dataset is used [9], [10]. The dataset contains multivariate time-series signals recorded from wearable sensors while people perform different activities. The dataset includes 19 activity classes and data from 8 subjects (p1–p8). The dataset splits each activity for each subject into short sequences, so the dataset has 9,120 sequences in total.

The sensors sit on five body locations (units): torso (T), arm (RA), left arm (LA), right leg (RL), and left leg (LL). The sensors on each unit have 9 channels. The 9 channels are -axial accelerometer (x,y z), tri-axial gyroscope (x,y z) and tri-axial magnetometer (x,y z). Therefore, each timestep has 45 features (5 units × 9 channels).

Each sample is a multichannel sequence of 125 time steps, so the input shape is:

- (T = 125, F = 45) per sequence
- Label: one of 19 activities
- Subject ID: used for subject-wise splitting

Missing values using np.isnan(...) are also checked and the dataset does not contain NaNs, so no imputation was needed.

III. MODEL DESCRIPTION

Because the input is multivariate time-series, the project tested several deep learning architectures that are commonly used for Human Activity Recognition. The project began with a baseline and then tested models that can capture temporal patterns better. The project also tested models that can use -sensor fusion. (Figure 1)

A. Baseline Model: 1D Convolutional Neural Network (1D-CNN)

The input is a multivariate sequence of length 125 with 45 channels. The network contains two temporal convolution blocks: a Conv1D layer with 64 filters (kernel size 5, ReLU, same padding). It is followed by Batch Normalization and MaxPooling1D (pool size 2), and a second Conv1D layer with 128 filters (kernel size 5, ReLU, same padding). It is also followed by Batch Normalization. GlobalAveragePooling1D is used to aggregate temporal features into a fixed-length representation. After GlobalAveragePooling1D, a layer with 64 units and ReLU activation is placed. Then Dropout with a rate of 0.6 is added after the layer. The final classification layer is a Softmax layer, with 19 units. The model has about 65k parameters.

B. Multi-Sensor Fusion Model

The fusion model takes the same input shape (125, 45), but instead of processing all 45 channels together, it splits the features into five 9-channel groups corresponding to the five body sensor units (Torso, Right Arm, Left Arm, Right Leg, Left Leg). This creates five parallel CNN branches (one per body location).

Each branch receives a (125, 9) sequence and the branch processes the sequence through two convolution blocks: a Conv1D layer with 64 filters (kernel size 5, ReLU, same padding) followed by Batch Normalization and MaxPooling1D (pool size 2), and a second Conv1D layer with 128 filters (kernel size 5, ReLU, same padding). It is also followed by Batch Normalization. Then, GlobalAveragePooling1D is applied in each branch to compress the temporal dimension into a fixed-length feature vector (128 features per branch).

The five branch outputs are then concatenated into a single vector (5 × 128 = 640 features). This fused representation is then fed into a Dense layer with 128 units and uses ReLU activation and Dropout (0.5). Finally, the model uses a 19-unit Softmax output layer to predict the activity class. The fusion model has about 308k parameters, which is larger than the baseline because it learns separate feature extractors for each body unit before combining them.

C. Temporal Convolutional Network(TCN) Model

The input is a multivariate sequence of 125 length with 45 channels. The TCN model is built from stacked temporal convolution blocks that use dilated Conv1D layers to capture short and long temporal dependencies without down sampling. Each residual block contains two Conv1D layers with 64 filters, a kernel size of 5 and same padding. The Conv1D layers use dilation rates of 1, 2 4 8 and 16 and are followed by the Batch Normalization, the ReLU activation and the Dropout, with a rate of 0.2. A residual connection is applied in each block. It uses a 1×1 Conv1D projection when needed to stabilize training and allow deeper temporal modeling. After the dilated blocks, GlobalAveragePooling1D aggregates features over time into a fixed-length representation. Its output goes into a Dense layer with 128 units (ReLU) and Dropout (0.5). The final classification layer produces a 19-unit Softmax output. The model has about 215k parameters.

- TCN tuned V1

In the tuned version (TCN v1) the model capacity is increased by using 128 filters of the smaller baseline setting. The kernel size = 5 and the dilation schedule (1, 2, 4, 8, 16) are kept the same. Also, regularization by setting dropout = 0.1 is reduced. Training was done with Adam (learning rate = 3e−4) and controlled using EarlyStopping and ReduceLROnPlateau to avoid overfitting and to stabilize convergence.

- TCN tuned V2

The second tuned version, TCN v2 uses 128 filters. The tuned version keeps the same optimizer and training strategy. The tuned version changes the temporal receptive field by using a smaller kernel size of 3 and a wider dilation schedule of 1, 2 4 8 16 32. The tuned version aims to capture longer-range temporal dependencies with a larger effective receptive field. The second tuned version also keeps dropout at 0.1. Uses the same early stopping and learning rate reduction strategy..

D. CNN + BiLSTM Model

The CNN+BiLSTM [11] model uses the input format as the baseline, that is a multivariate time series with shape (125, 45). The CNN feature extractor is the part of the network that contains two temporal convolution blocks. The first block applies a Conv1D layer with 64 filters (kernel size 5, ReLU, same padding), and it is followed by Batch Normalization and MaxPooling1D (pool size 2) to reduce the temporal length. The second block applies a Conv1D layer with 128 filters (kernel size 5, ReLU, same padding), followed by Batch Normalization and another MaxPooling1D (pool size 2). This produces a shorter sequence of learned features with shape approximately (31, 128).

After the CNN blocks the system adds a Bidirectional LSTM with 64 units. The Bidirectional LSTM looks at the remaining sequence in both backward directions. The BiLSTM creates a fixed length representation. The fixed length representation goes into a layer with 64 units and ReLU activation. The Dense layer also uses Dropout set to 0.5 for regularization. The final layer is a Softmax layer with 19 units for activity classification. The model contains about 165k parameters. The model is bigger, than the baseline because the recurrent BiLSTM layer adds parameters.

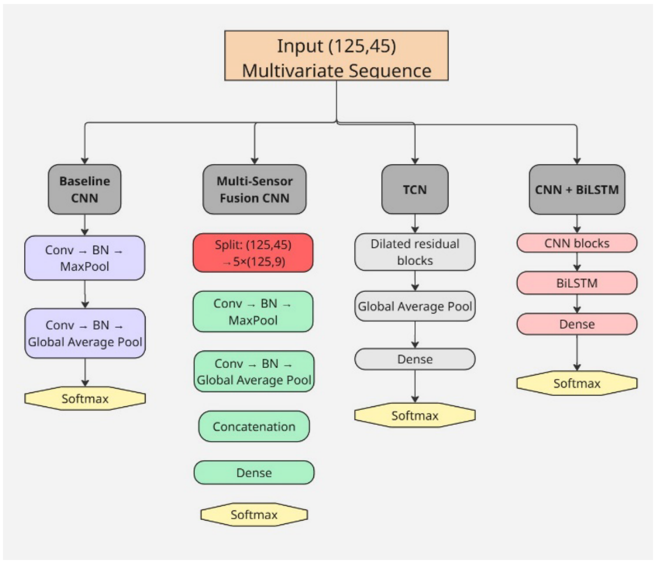


Figure 1: Schematic diagram of proposed neural network architectures for multivariate sequence classification

IV. EXPERIMENTAL SETUP

a) Data Split (Subject-wise)

To evaluate how well the models generalize to unseen people, a subject-wise split is used. Subjects 1–6 were used for training/validation, and subjects 7–8 were used as the test set. This avoids information leakage because the same person does not appear in both training and test data.

From the training subjects (1–6), a validation set by splitting 20% of the training data using a stratified split is used (keeping the class distribution similar).

This resulted in:

- Train_full: 6,840 samples
- Train: 5,472 samples
- Validation: 1,368 samples
- Test: 2,280 samples

Each sample is a multivariate time-series with shape (125, 45).

b) Preprocessing

Before training, the dataset was checked for missing values (NaNs). No missing values were found, so no imputation was needed.

We applied standardization (z score scaling) for normalization using the training set statistics. Specifically we fitted a StandardScaler on the training data. Then we applied the StandardScaler to the validation set and the test set. Applying the StandardScaler this way makes sure that the test set does not influence preprocessing.

c) Training Procedure

All models were trained for multi-class classification using:

- Loss: Sparse Categorical Cross-Entropy (labels are integers 0–18)
- Optimizer: Adam
- Batch size: 64

The main training settings per model were:

- Baseline 1D-CNN: learning rate = 5e-4, epochs = 100
- TCN (base): learning rate = 5e-4, epochs = 100
- Fusion CNN (5-branch): learning rate = 5e-4, epochs = 100
- CNN + BiLSTM: learning rate = 3e-4, epochs = 200, with EarlyStopping (patience = 10)
- Tuned TCN v1 / v2: trained up to 200 epochs using EarlyStopping and ReduceLROnPlateau to control overfitting and automatically reduce the learning rate when validation performance stopped improving.

d) Evaluation Metrics

Model performance was evaluated on the held-out test subjects (7–8). Since this is a 19-class problem, it reported:

- Test Accuracy
- Macro Precision / Macro Recall / Macro F1 (important because it weighs each class equally)
- Confusion Matrix (counts)

- Top confusions (largest off-diagonal errors)

In addition, we reported each model's number of parameters to compare model complexity.

The code used for preprocessing, training, and evaluation is available online at: https://github.com/Faezeh-sb/ANN2-HAR-Multivariate-TimeSeries

V. RESULTS

This section reports the final test performance on the held-out subjects (p7–p8). This project compared all models using test accuracy, macro precision/recall/F1, and analyzed the training curves and the main confusions (off-diagonal errors).

Overall Performance (Test Set)

Based on the final evaluation summary (Table 1), the Fusion model achieved the best overall performance with a test accuracy of 0.8715 and macro F1 of 0.87. This suggests that splitting sensors by body location and fusing features at a later stage helps generalization across unseen subjects.

The second best model is TCN v1. TCN v1 gets an accuracy of 0.8461 and a macro F1 of 0.8473. The tuned version of the base TCN beats the base TCN. It improves the results noticeably. It shows that increasing model capacity (filters) and adjusting regularization helps models.

The Baseline 1D-CNN did well (accuracy = 0.8399, F1 = 0.8431) even though the Baseline 1D-CNN has the smallest number of parameters (65k). So that makes the Baseline 1D-CNN a simple but solid reference model.

The base TCN performance with a test accuracy of 0.8311 and macro F1 = 0.8332. So, it did not outperform Fusion or TCN v1 or baseline CNN. Finally, CNN+BiLSTM (accuracy = 0.8162, macro F1 = 0.8188) was below the baseline and the base TCN, and the TCN v2 was the weakest model (accuracy = 0.8105, macro F1 = 0.8159.

Overall, in this project, the best generalization came from multi-sensor fusion, followed by the stronger tuned TCN. In terms of complexity (parameters), Fusion (308k) is larger than the baseline, while the tuned TCN (TCNv1) variants are the largest models (Table 1). So, the best performance also comes with higher model size, but the baseline remains competitive given its small capacity.

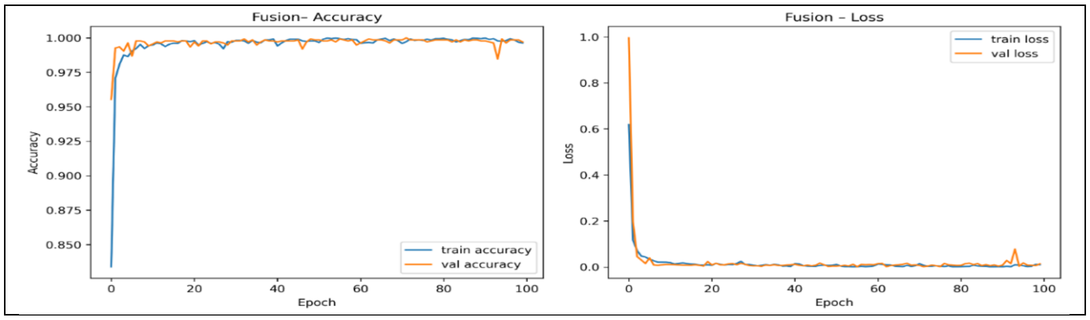
Table 1: Evaluation results of the implemented models

Model	*Accuracy*	*Precision (Macro)*	*Recall (Macro)*	*F1 (Macro)*	*Precision (Weighted)*	*Recall (Weighted)*	*F1 (Weighted)*	*#Params*
Fusion	0.8715	0.9055	0.8715	0.87	0.9055	0.8715	0.87	308499
TCN v1	0.8461	0.892	0.8461	0.8473	0.892	0.8461	0.8473	797331
Baseline	0.8399	0.8981	0.8399	0.8431	0.8981	0.8399	0.8431	65811
TCN	0.8311	0.8785	0.8311	0.8332	0.8785	0.8311	0.8332	215635
CNN+BiLSTM	0.8162	0.8814	0.8162	0.8188	0.8814	0.8162	0.8188	164627
TCN v2	0.8105	0.8725	0.8105	0.8159	0.8725	0.8105	0.8159	590483

Training Behavior (Accuracy/Loss Curves)

From the learning curves (Figure.2), both Baseline CNN and Fusion reach very high training/validation accuracy quickly, and their losses drop close to zero after the early epochs. This indicates that the models can fit the training subjects easily. However, the test accuracy is lower than the near-perfect training accuracy. Actually it is expected because the test split is subject-wise (new people), and the movement style can change across subjects.

For the tuned models (TCN v1, TCN v2) and CNN+BiLSTM, the curves show fast convergence and stable behavior overall. In some runs, small spikes appear in validation loss even when accuracy stays high (this can happen when the model becomes very confident on most samples, but a few validation samples remain hard). EarlyStopping and ReduceLROnPlateau stop training soon as validation performance stops improving, which helps reduce overfitting.



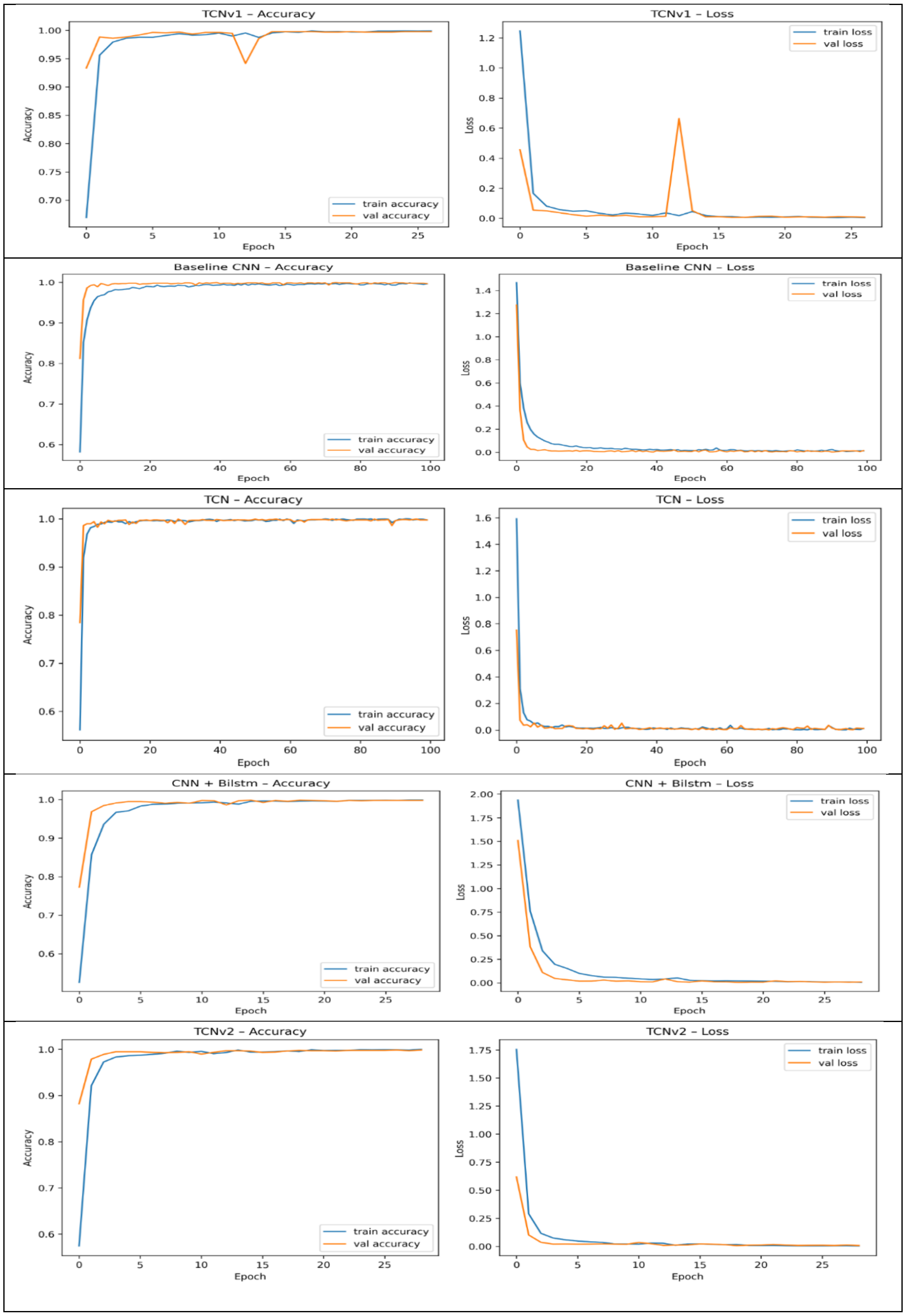


Figure 2: Performance curves of the proposed models. The figure provides a side-by-side comparison of Loss and Accuracy curves for all six experimental setups

- Confusion Analysis (Most Frequent Errors)

To understand where the models fail, this project extracted the strongest off-diagonal confusions (Table 2). Across multiple models, the most common errors happen between activities that are naturally similar:

Treadmill walking variants were confused with each other (e.g., walking on a slope treadmill vs walking on a flat treadmill at 4 km/h). This makes sense because the motion pattern is very similar and only differs slightly in incline.

Elevator-related classes were also a frequent confusion source. For example, standing (A1) was often predicted as moving around in an elevator (A7) and standing still in an elevator (A6) was confusing with moving in an elevator (A7). These activities share limited motion. The only real difference is a few body shifts.

Some models showed confusion between a walking class (e.g., walking in a parking lot (A8)) and a sports activity (e.g., playing basketball (A18)). These activities share limited motion and may differ mostly in subtle body shifts.

For the base TCN, one major confusion was sitting (A0) being predicted as cycling (A14). That tells us that the base temporal model does not separate low-motion repeating patterns well as the tuned versions or the fusion.

Overall, the confusion patterns are consistent with the dataset: activities with similar intensity and similar sensor signatures are the hardest to separate. The Fusion model improves the final accuracy, but the remaining errors are still mostly between near activities (especially treadmill and elevator-related classes), which is reasonable.

Table 2: Top 2 Confusion for all models

MODEL	TRUE LABEL	PRED LABEL	TRUE ACTIVITY	PRED ACTIVITY	COUNT (SAMPLES)
BASELINE 1D-CNN	10	9	walking with slop on a treadmill with a speed of 4 km/h(A10)	walking on a flat treadmill with a speed of 4 km/h(A9)	60
BASELINE 1D-CNN	5	7	descending stairs (A5)	moving around in an elevator (A7)	60
CNN+BILSTM	8	18	walking in a parking lot (A8)	playing basketball (A18)	60
CNN+BILSTM	10	9	walking with slop on a treadmill with a speed of 4 km/h(A10)	walking on a flat treadmill with a speed of 4 km/h(A9)	59
FUSION (5-BRANCH)	8	18	walking in a parking lot (A8)	playing basketball (A18)	60
FUSION (5-BRANCH)	1	7	standing (A1)	moving around in an elevator (A7)	60
TCN	0	14	sitting (A0)	cycling on an exercise bike in horizontal (A14)	60
TCN	1	7	standing (A1)	moving around in an elevator (A7)	60
TCN V1	1	7	standing (A1)	moving around in an elevator (A7)	60
TCN V1	10	9	walking with slop on a treadmill with a speed of 4 km/h(A10)	walking on a flat treadmill with a speed of 4 km/h(A9)	59
TCN V2	1	7	standing (A1)	moving around in an elevator (A7)	60
TCN V2	6	7	standing in an elevator still (A6)	moving around in an elevator (A7)	58

I. DISCUSION

Overall, the experiments show that the models can learn meaningful patterns from the wearable signals, even with a subject-wise split (train on subjects 1–6, test on 7–8). In our case, the multi-sensor Fusion CNN achieved the best overall performance. It makes sense because this dataset is collected from five body locations, and treating each body unit as its own branch helps the network learn location-specific motion cues before combining them. This idea is also consistent with how this dataset is described in other works (multiple sensor units across the body and multiple modalities like accelerometer/gyroscope/magnetometer).[12] [13]

A second observation looks at model complexity and real generalization. Some models (especially the tuned TCN versions) have many more parameters, and we can see one of them outperformed the simpler models on the unseen-subject test set, while the other did not. This suggests the main bottleneck is not capacity, but how well the model handles inter-subject variation (different people performing the same activity slightly differently). This challenge is actually emphasized in the literature for this dataset too [14], since the recordings come from multiple subjects and activities that can overlap in motion style.[14], [15] [16]

Many recent HAR papers use CNN-based feature extraction and different forms of fusion or temporal modeling (e.g., parallel CNN branches, deeper temporal blocks).[17] Our findings support that direction: explicit multi-sensor fusion was the most effective design choice in this project. For future work, the most useful next steps would be: (i) report stability across multiple runs, (ii) try stronger cross-subject evaluation, and (iii) test improved fusion (attention/weighted fusion) or augmentation to reduce sensitivity to subject differences.

Finally, there are clear limitations. The dataset is relatively small (only 8 subjects), so results can change between runs due to randomness (initial weights, minibatch order, etc.). Some activities are naturally similar, so confusion is expected even with good models; this is a dataset difficulty issue, not only a model issue. We only used one subject split (1–6 vs. 7–8). A stronger evaluation would be leave-one-subject-out or multiple splits to report mean ± std. We did not do heavy hyperparameter search, so some models may be under-tuned compared to what is possible.

All reported numbers in the Results section are taken from the final run that we decided to keep.

I. CONCLUSION

In this project, several deep learning models for multivariate wearable-sensor activity recognition were evaluated. It used a subject-wise split (train on subjects 1–6, test on 7–8). Overall, the models were able to learn useful patterns from the raw signals and achieve good generalization to unseen subjects. The best results came from the multi-sensor Fusion CNN, which suggests that explicitly separating body locations and then fusing their features is a strong approach for this dataset.

Even though some models were larger, higher capacity did not always guarantee better performance, so generalization across different people seems to be the main challenge. For future work, the most useful improvements would be using more robust evaluation (multiple splits / LOSO) and testing stronger fusion ideas (e.g., attention-based fusion) to reduce sensitivity to subject differences.

References

[1] S. Zhang *et al.*, "Deep Learning in Human Activity Recognition with Wearable Sensors: A Review on Advances," *Sensors*, vol. 22, no. 4, p. 1476, Feb. 2022, doi: 10.3390/s22041476.

[2] M. Kaseris, I. Kostavelis, and S. Malassiotis, "A Comprehensive Survey on Deep Learning Methods in Human Activity Recognition," *Mach. Learn. Knowl. Extr.*, vol. 6, no. 2, pp. 842–876, Apr. 2024, doi: 10.3390/make6020040.

[3] A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," *ACM Comput. Surv.*, vol. 46, no. 3, pp. 1–33, Jan. 2014, doi: 10.1145/2499621.

[4] J. B. Yang, M. N. Nguyen, P. P. San, X. L. Li, and S. Krishnaswamy, "Deep Convolutional Neural Networks on Multichannel Time Series for Human Activity Recognition".

[5] M. M. Islam, S. Nooruddin, F. Karray, and G. Muhammad, "Human Activity Recognition Using Tools of Convolutional Neural Networks: A State of the Art Review, Data Sets, Challenges and Future Prospects," *Comput. Biol. Med.*, vol. 149, p. 106060, Oct. 2022, doi: 10.1016/j.compbiomed.2022.106060.

[6] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[7] F. Ordóñez and D. Roggen, "Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition," *Sensors*, vol. 16, no. 1, p. 115, Jan. 2016, doi: 10.3390/s16010115.

[8] S. Bai, J. Z. Kolter, and V. Koltun, "An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling," Apr. 19, 2018, *arXiv*: arXiv:1803.01271. doi: 10.48550/arXiv.1803.01271.

[9] K. A. Billur Barshan, "Daily and Sports Activities." UCI Machine Learning Repository, 2010. doi: 10.24432/C5C59F.

[10] K. Altun, B. Barshan, and O. Tunçel, "Comparative study on classifying human activities with miniature inertial and magnetic sensors," *Pattern Recognit.*, vol. 43, no. 10, pp. 3605–3620, Oct. 2010, doi: 10.1016/j.patcog.2010.04.019.

[11] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997, doi: 10.1109/78.650093.

[12] B. Barshan and M. C. Yuksek, "Recognizing Daily and Sports Activities in Two Open Source Machine Learning Environments Using Body-Worn Sensor Units," *Comput. J.*, vol. 57, no. 11, pp. 1649–1667, Nov. 2014, doi: 10.1093/comjnl/bxt075.

[13] B. Barshan and A. Yurtman, "Classifying Daily and Sports Activities Invariantly to the Positioning of Wearable Motion Sensor Units," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 4801–4815, Jun. 2020, doi: 10.1109/JIOT.2020.2969840.

[14] B. Barshan and A. Yurtman, "Investigating Inter-Subject and Inter-Activity Variations in Activity Recognition Using Wearable Motion Sensors," *Comput. J.*, vol. 59, no. 9, pp. 1345–1362, Sep. 2016, doi: 10.1093/comjnl/bxv093.

[15] A. Dehghani, T. Glatard, and E. Shihab, "Subject Cross Validation in Human Activity Recognition," Apr. 09, 2019, *arXiv*: arXiv:1904.02666. doi: 10.48550/arXiv.1904.02666.

[16] C. F. S. Leite and Y. Xiao, "Improving Cross-Subject Activity Recognition via Adversarial Learning," *IEEE Access*, vol. 8, pp. 90542–90554, 2020, doi: 10.1109/ACCESS.2020.2993818.

[17] H. Zhao, "A parallel CNN architecture for sport activity recognition based on minimal movement data," *Sci. Rep.*, vol. 14, no. 1, p. 31697, Dec. 2024, doi: 10.1038/s41598-024-81733-z.