PAMP2 Physical Activity Monitoring

Ken Lai, Quang Tran (Team 6)

Shiley-Marcos School of Engineering, University of San Diego

AAI-530: Data Analytics and Internet of Things

Prof. Roozbeh Sadeghian, PhD

Feb 14, 2026

Abstract

The PAMAP2 Physical Activity Monitoring dataset is a multi-modal sensor dataset created by Atilla Reiss (2012) to support research in human activity recognition (HAR) and related machine learning applications. The dataset was collected from 9 healthy subjects performing a range of daily and exercise activities while wearing Inertial Measurement Units (IMUs) and heart rate monitors. Sensor streams include tri-axial accelerometers, gyroscope, magnetometer, and heart-rate data collected at high sampling frequencies and wirelessly transmitted to a companion computing unit. This dataset presents real-world challenges such as wireless packet loss and varying activity intensities, making it ideal for evaluating deep learning and time-series modeling approaches. The PAMAP2 data is widely used in HAR benchmarking, activity segmentation, and wearable computing research due to its broad feature set, labeled activities, and extensive documentation.

Keywords: machine learning, deep learning, random forest classification, neural networks

Introduction

Advances in wearable sensor technology and machine learning have enabled highly accurate recognition of human activities in real-time use cases. The PAMAP2 Physical Activity Monitoring dataset was created to provide a high-resolution, multimodal benchmark for developing and evaluating activity recognition algorithms. Collected from nine healthy subjects performing twelve diverse physical activities — including walking, cycling, household chores, and dynamic exercises — the dataset combines inertial signals from IMUs and continuous heart rate measurements. The integrated use of multiple sensors and frequent sampling rates yields a good representation of human motion dynamics. Additionally, the wireless nature of data acquisition introduces transmission variability, mirroring practical deployment conditions for Internet of Things (IoT) and wearable systems.

Hosted by the UCI Machine Learning Repository and accompanied by voluminous metadata, the PAMAP2 dataset has become foundational in human activity recognition research. Its comprehensive labeling schema and real-world complexities support the benchmarking of traditional classification methods as well as modern deep learning. As researchers work to build robust HAR systems for health monitoring, elder care, sports analytics, and ubiquitous computing, PAMAP2 serves as a baseline dataset for algorithm development and comparative analysis.

IoT System Design

Our proposed AIoT system is designed to collect, transmit, process, and analyze multi-modal physiological and motion data from wearable devices to enable reliable physical activity monitoring. The system leverages a combination of on-body sensors, light-weight edge processing, wireless connectivity, cloud-based data management, and scalable machine learning workflows. Its architecture reflects the requirements of continuous sensing, heterogeneous sampling rates, and real-time inference common in wearable IoT applications.

- **Sensors**: The system is comprised of three wearable Inertial Measurement Units (IMUs) positioned on the **hand**, **chest**, and **ankle**. Each IMU includes multiple sensing modalities with a sampling frequency of 100 Hz:
  - 3-axis accelerometer (±16 g)

- 3-axis accelerometer (±6 g)
- 3-axis gyroscope (±1,200°/s)
- 3-axis magnetometer

At the same time, a **heart rate monitor** operating near 9 Hz provides physiological data to complement motion signals. The combination of high-frequency IMU data with low-frequency heart-rate measurements supports modeling of body movement, exertion level, and transitions between various activities. This multi-modal design reflects a typical wearable IoT configuration where sensors enhance model accuracy and robustness.

- **Edge Processing**: Limited edge processing is performed on the wearable nodes to prepare data for wireless transmission. Because the wearable devices are resource-constrained and battery-powered, no complex machine learning inference is executed on the edge. Instead, the edge layer focuses on efficient and low-latency data acquisition, preserving raw sensor quality for downstream analysis.
- **Network Architecture**: The wearable sensors communicate wirelessly with a base station – typically a mobile receiver or laptop – using a short-range radio connection. Each IMU packet is tagged with a system timestamp to support multi-sensor alignment, particularly important given the differing sampling frequencies across devices. The data stream is forwarded through a local network to cloud storage for long-term retention and processing.
- **Data Storage and Processing**: Collected data are stored in the cloud capable of handling high-frequency time-series sensor streams. Because the combined dataset from all subjects exceeds several million samples, cloud storage provides the reliability and scalability necessary for long-term retention, access by multiple analytic pipelines, and integration with machine learning frameworks. Preprocessing tasks such as cleaning missing values, generating derived features, and time alignment are performed using cloud-based compute resources. Feature engineering and model training occur in this environment as well, leveraging scalable processing engines to support larger datasets and more complex models.
- **Scalability**: The architecture is designed to accommodate additional subjects, more sensor modalities, or higher sampling rates without requiring significant changes. The cloud-based pipeline enables horizontal scaling of both compute and storage resources, while the modular sensor architecture allows new wearable devices to be integrated with minimal modification. As model complexity increases—such as incorporating deep

learning models like convolutional or recurrent neural networks—the system can seamlessly leverage more powerful compute instances.

- **Use Cases**: This AIoT system supports numerous applications involving activity recognition and health or performance monitoring. In fitness and wellness scenarios, the system can classify user activities, estimate exertion levels, and detect transitions such as walking, jogging, or cycling. In healthcare contexts, the system can evaluate mobility patterns, monitor recovery progress, or detect abnormal gait behavior. Industrial safety applications include real-time monitoring of worker fatigue, improper lifting techniques, or hazardous motion events.

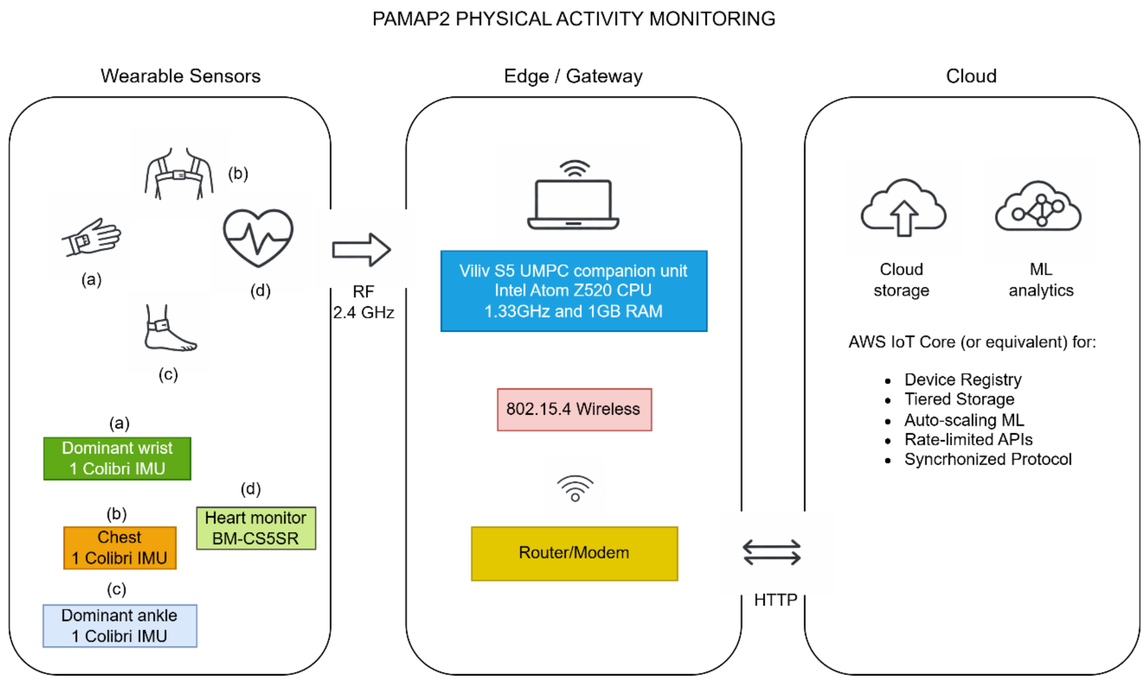

Figure 1. IoT diagram of the project.

PAMAP2 monitoring for activity recognition, health monitoring, or anomaly detection

Dataset and Preprocessing

Preprocessing began with merging the nine subject files into a single data frame. Column names were assigned manually based on the dataset documentation to ensure correct alignment of sensor variables. To preserve subject identity for downstream modeling and generalization testing, a subject_id variable was added to each file prior to concatenation. Once merged, the dataset contained approximately 2.87 million observations across 56 variables, including timestamp, activity labels, heart rate, and multi-axis sensor measurements from each body location. Given the dataset's massive size, memory-efficient loading and storage strategies were employed, including saving intermediate results in Parquet format to improve read and write performance during iterative analysis.

	Metric	Value
0	Total Observations	2872533
1	Total Features	59
2	Number of Subjects	9
3	Number of Activities	13
4	Sampling Frequency (Hz)	100

Table 1: Dataset Summary Statistics

Following integration, extensive datatype correction and cleaning were performed. Several sensor columns were initially interpreted as object types due to formatting inconsistencies and required explicit conversion to numeric formats. All continuous sensor measurements were cast to floating-point types to ensure compatibility with statistical analysis and machine learning libraries. The timestamp column was converted into a proper datetime format and used to sort observations chronologically within each subject.

Comprehensive analysis was performed to check for completeness. Most sensor variables had minimal values missing; however, a small number of orientation-related fields contained either sparse or entirely missing values. Columns that were fully missing were excluded from further modeling. For intermittent missing sensor readings, imputation strategies appropriate for time-series data were applied, such as forward-filling or zero-imputation depending on the semantics of the feature. Visualizations of missing data patterns confirmed that missingness was not concentrated within specific time intervals or subjects.

	0
ankle_orient_4	2872533
activity_id_2	2610265
hand_acc16_y	13141
hand_acc16_x	13141
heart_rate	13141
hand_acc16_z	13141
hand_acc6_z	13141
hand_acc6_y	13141
hand_temp	13141
hand_mag_z	13141
hand_orient_1	13141
hand_orient_2	13141
hand_gyro_x	13141
hand_gyro_y	13141
hand_gyro_z	13141

dtype: int64

Figure 2: Missing Data Visualization

Exploratory data analysis (EDA) was then conducted to understand the dataset's structural properties. Activity label distribution revealed significant class imbalance, with activity ID 0 appearing far more frequently than other labeled activities. This imbalance suggests that classification models trained directly on the dataset may bias toward the dominant class without corrective techniques such as weighting or resampling.

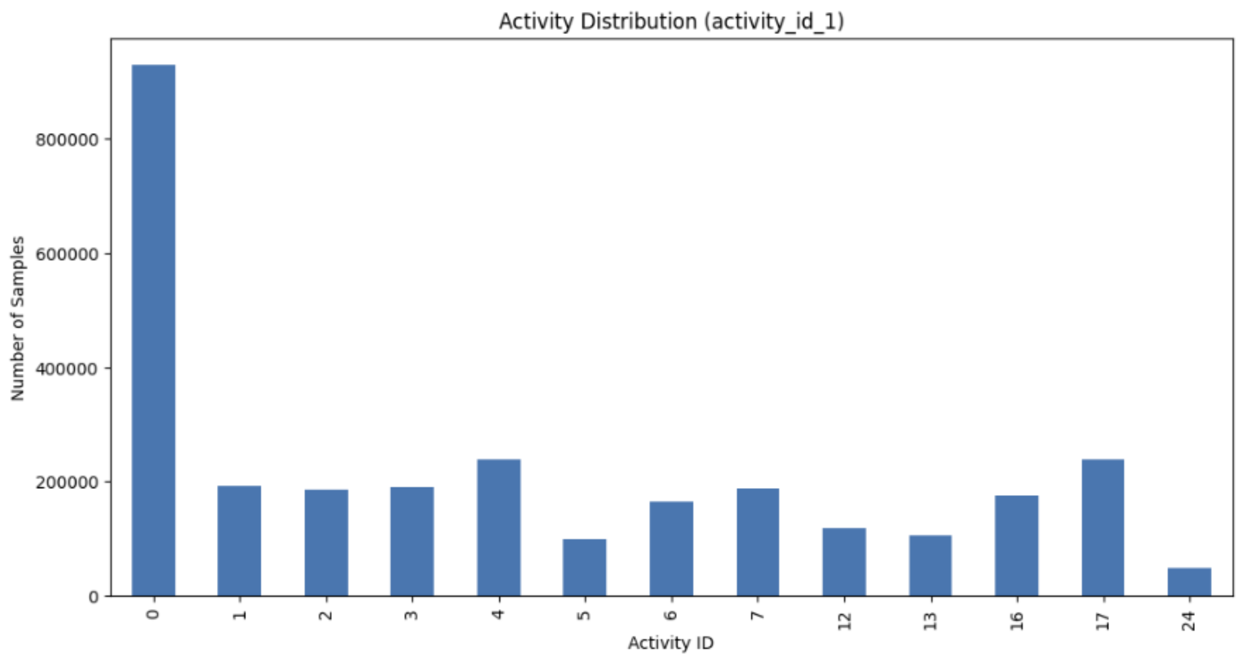

Figure 3: Activity Distribution Bar Chart

Correlation analysis provided further insight into relationships between modalities. A Spearman correlation heatmap demonstrated strong positive correlations among accelerometer magnitudes across body locations. Moderate correlations were observed between hand movement intensity and heart rate, indicating that upper-body activity corresponds with increased physiological load. On the other hand, ankle motion showed weaker correlation with heart rate, suggesting that certain lower-body movements may not affect cardiovascular response.

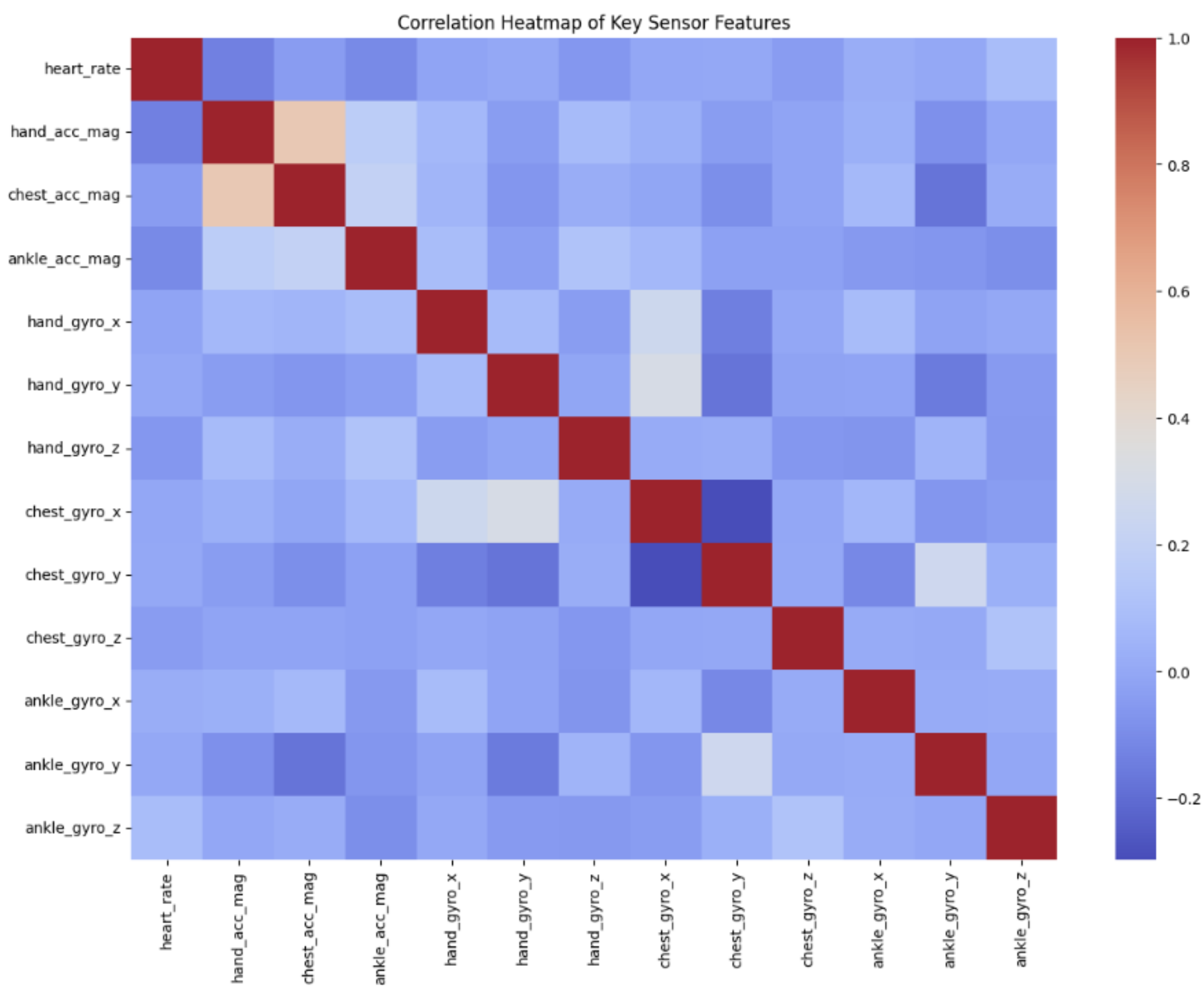

Figure 4: Correlation Heatmap of Key Sensor Features

Because the dataset contains records from multiple individuals, subject-level variability was examined to assess potential drift and generalization challenges. Baseline heart rate differences across participants were noticeable, and movement magnitudes varied even during similar activities. This variability highlights the importance of preventing subject leakage when splitting the dataset. A subject-wise 80/20 split was therefore implemented, ensuring that entire subjects were assigned either to the training or test set.

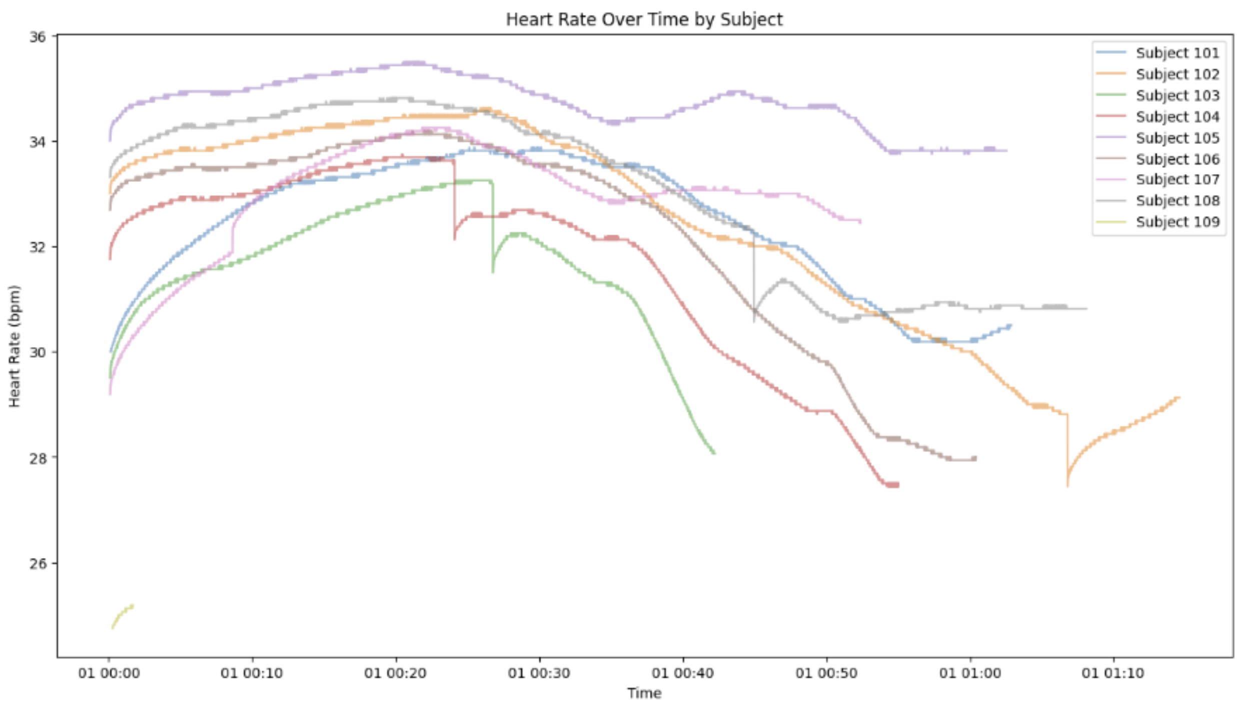

Figure 5: Heart Rate Drift Per Subject Visualization

To prepare the dataset for sequence-based machine learning models, particularly Long Short-Term Memory (LSTM), sliding-window sequences were constructed from the chronologically sorted data. Each training sample consisted of a fixed-length sequence of sensor readings (e.g., 30 timesteps) used to predict a future value at a defined prediction horizon (e.g., 5 timesteps ahead). The resulting input arrays were structured in three dimensions: number of samples, sequence length, and number of features. For streaming scenarios, padding techniques were used to accommodate sequences shorter than the required window length, ensuring consistent input dimensionality for recurrent neural networks.

Overall, the dataset transformation process converted raw sensor recordings into a structured, analysis-ready format suitable for various machine learning applications. The preprocessing included file integration, datatype correction, timestamp alignment, missing value treatment, correlation and drift analysis, subject-wise dataset splitting, and sequence generation. These steps ensured data integrity, preserved temporal structure, minimized leakage, and established a robust foundation for modeling and analysis.

Machine Learning Classification

The initial phase of the project focused on the activity classification problem from PAMAP2 wearable sensor data. To reduce label noise and avoid difficult-to-learn transitions, we retained the stable activity labels and removed activity IDs 0 and 24. The data was then split by subject to prevent information leakage between users, with approximately 20% of the subjects selected as the test set and the remaining subjects as the training set. To convert time-series into supervised training data, we used a 10-second sliding window with 50% overlap and ensured each window contained only a single constant-label segment to avoid overlapping labels within the same window. Each window had its time-domain and frequency-domain features extracted, including statistics such as mean, standard deviation, min, max, median, IQR, energy, jerk, and band-force energy from the FFT, thus creating a fixed-feature vector for each window.

To evaluate the model's fairness across subjects, the notebook used GroupKFold subject-specific cross-validation. A comparison of classical models showed that ExtraTrees was the strongest in cross-validation, with macro F1 and balanced accuracy means of approximately 0.95 and 0.946, respectively, surpassing RandomForest, HistGradientBoosting, Logistic Regression, and LinearSVC. When retrained on the training set and evaluated on the subject-specific test set, ExtraTrees achieved a macro F1 mean of approximately 0.948, a balanced accuracy mean of approximately 0.947, and an accuracy mean of approximately 0.9418. Additionally, the notebook fine-tuned LinearSVC using RandomizedSearchCV, achieving macro F1 of approximately 0.949, balanced accuracy of approximately 0.947, and accuracy of approximately 0.946. Thus, with window-specific data, the two best machine learning options were ExtraTrees and the refined LinearSVC, with LinearSVC slightly better in macro F1 on the test set, while ExtraTrees clearly led in subject-specific cross-validation results. The confusion matrix in the notebook showed that most classes were well-discriminated, but some classes were more easily confused due to similar motion or close signal amplitudes.

```
best model: ExtraTrees
TEST macro F1: 0.9476640061215146
TEST balanced acc: 0.946905599798313
              precision    recall  f1-score   support

           1     1.0000    0.9778    0.9888        90
           2     0.9610    0.8409    0.8970        88
           3     0.9881    0.8557    0.9171        97
           4     0.8636    1.0000    0.9268       114
           5     1.0000    1.0000    1.0000        61
           6     0.9888    1.0000    0.9944        88
           7     1.0000    0.8273    0.9055       110
          12     0.9492    1.0000    0.9739        56
          13     1.0000    0.9574    0.9783        47
          16     0.8602    0.9877    0.9195        81
          17     0.8811    0.9692    0.9231       130

    accuracy                         0.9418       962
   macro avg     0.9538    0.9469    0.9477       962
weighted avg     0.9473    0.9418    0.9414       962
```

Figure 6: ExtraTrees Model Summary

Deep Learning CNN Model for Operational Classification

To test the ability to learn directly from sequences, we implemented a one-dimensional CNN model on time-based window data. The model was trained for 100 epochs, and macro F1 and balanced accuracy were monitored on the test set at each epoch. In the final results, the CNN achieved a test macro F1 of approximately 0.93, a test balanced accuracy of approximately 0.92, and an accuracy of approximately 0.93. Compared to classical window-based feature models, CNN still had a lower macro F1 of approximately 0.02 compared to ExtraTrees and LinearSVC. This is appropriate for the project context because classical models are supported by strong feature sets in the time and frequency domains, whereas CNNs require more data and better architectural design and optimization to overcome the need for manual feature modeling in this application. Nevertheless, CNN still yielded good results and is clear evidence for the direction of deep learning on windowed time-series data.

	Model	Validation BalancedAcc	Validation MacroF1	Test Accuracy	Test BalancedAcc	Test MacroF1
0	ExtraTrees	0.9462	0.9496	0.9418	0.9469	0.9477
1	CNN 1D	nan	nan	0.9314	0.9235	0.9307
2	LinearSVC	0.8470	0.8188	0.9179	0.9140	0.9219
3	LogReg	0.8762	0.8532	0.9168	0.9104	0.9137
4	RandomForest	0.9338	0.9172	0.9044	0.9129	0.9075
5	HistGradientBoosting	0.8726	0.8619	0.9033	0.9058	0.9074

Figure 7: Comparison of CNN vs other Machine Learning Models

Visualization Using Tableau Public

After obtaining a reliable activity classification model, the next step was to provide a visualization in Tableau Public. A Tableau dashboard allows users to select subjects and view activity duration, the most recent activity, sensor signal histograms over time, and predicted activity labels. With the window-based representation, prediction labels are assigned based on the window's end time and then projected down to the one-second level to fit the timeline in

Tableau. This method transforms the prediction pipeline from the notebook into tabular data that can be directly dragged and dropped into the dashboard.

Figure 8: Tableau Dashboard

Future Model Improvements

There exist many ways to improve accuracy, reduce confusion between closely related layers, and enhance large-scale deployment capabilities. One way involves expanding the dataset by increasing the number of subjects, activity diversity, and recording duration, as models perform better when they see a wider variety of movements. As data volumes increase, the pipeline optimizes performance by loading and processing multiple files in parallel, extracting features in parallel with ‘joblib’, and training the model with multi-threading, while also leveraging GPUs to accelerate deep learning model training. Additionally, data augmentation techniques such as sensor-based noise generation or SMOTE oversampling for low-sampling-rate activities can reduce class imbalance, thereby significantly improving F1 macro.

Still another improvement would be tighter control over data leakage by subject-based separation from the outset, before normalization and feature extraction. Normalization of the entire dataset before splitting the training and test could allow statistical information from the test set to "leak" into the training set. Maintaining subject-based GroupKFold in cross-validation also helps ensure stability and accurately reflects generalization capabilities for new users.

In terms of features, more informative time-domain features, such as RMS, kurtosis, and skewness, can be added to differentiate the jerk intensity and signal distribution patterns across activities. In the frequency domain, features like dominant frequency, spectral centroid, spectral entropy, and multiple power bands often help differentiate activities with different rhythms. Furthermore, for gait-related problems, gait features such as step-time mean, step-time coefficient of variation, and cadence are useful for identifying changes in movement or functional decline. One advanced approach is to measure phase synchronization between sensors using Hilbert transforms to identify complex movements when multiple body parts coordinate. This can improve the accuracy of classes prone to confusion due to similar amplitudes but different phases of movement.

Finally, Deep Learning models can be improved by better exploiting long-term dependence. Instead of relying solely on the static features of the window, LSTM or hybrid CNN LSTM models can be used to learn feature sequences over time, thereby capturing continuous and transitional movement patterns. This approach is particularly useful for activities with rhythmic or gradually changing structures over time and can help increase accuracy in classes that are difficult to distinguish using only short windows.

Conclusion

Our results show that activity classification from wearable sensor data can achieve very high accuracy when using a subject-based data-splitting process to reflect generalization to new users, combined with time-series representation via sliding windows and feature extraction in both the time and frequency domains. Classical window-based feature models such as ExtraTrees and LinearSVC, when refined, achieve a macro F1 of approximately 0.95 on the subject-based test set, demonstrating the pipeline's stability and practical effectiveness. The one-dimensional CNN model also yielded strong results, achieving a macro F1 of around 0.93, indicating that direct learning from the series has potential and can be further improved through data scaling, architecture optimization, and refined training strategies. Overall, this approach is suitable for implementing activity tracking and analysis in health and fitness applications and provides a foundation for developing advanced functionalities, such as predicting the next activity and detecting movement abnormalities early.

References

"PAMAP2 Physical Activity Monitoring." *UCI Machine Learning Repository*, https://archive.ics.uci.edu/dataset/231/pamap2+physical+activity+monitoring. Accessed 14 Feb. 2026.

Batt, S., & Kim, J. (2020). Learning Tableau A data visualization tool. Journal of Economic Education, 51(3–4), 317–328. https://doi.org/10.1080/00220485.2020.1804503

Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE Synthetic minority over sampling technique. Journal of Artificial Intelligence Research, 16, 321–357. https://doi.org/10.1613/jair.953

Cooley, J. W., & Tukey, J. W. (1965). An algorithm for the machine calculation of complex Fourier series. Mathematics of Computation, 19(90), 297–301. https://doi.org/10.1090/S0025-5718-1965-0178586-1

Cortes, C., & Vapnik, V. (1995). Support vector networks. Machine Learning, 20, 273–297. https://doi.org/10.1007/BF00994018

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. Machine Learning, 63, 3–42. https://doi.org/10.1007/s10994-006-6226-1

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., Fernández del Río, J., Wiebe, M., Peterson, P., & Oliphant, T. E. (2020). Array programming with NumPy. Nature, 585(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hunter, J. D. (2007). Matplotlib A 2D graphics environment. Computing in Science & Engineering, 9(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

McKinney, W. (2010). Data structures for statistical computing in Python. Proceedings of the 9th Python in Science Conference, 56–61.

Moya Rueda, F., Grzeszick, R., Fink, G. A., Feldhorst, S., & ten Hompel, M. (2018). Convolutional neural networks for human activity recognition using body worn sensors. Information, 9(2), 26.

Nguyen, B., Mengistu, B. M., & Bui, F. M. (2021). Trends in human activity recognition with focus on machine learning and deep learning approaches. Machine Learning with Applications, 6, 100200.

Ordóñez, F. J., & Roggen, D. (2016). Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition. Sensors, 16(1), 115.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., & Chintala, S. (2019). PyTorch An imperative style high performance deep learning library. Advances in Neural Information Processing Systems, 32.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit learn Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830.

Ravi, N., Dandekar, N., Mysore, P., & Littman, M. L. (2005). Activity recognition from accelerometer data. Proceedings of the AAAI Conference on Artificial Intelligence.

Reiss, A., & Stricker, D. (2012). Creating and benchmarking a new dataset for physical activity monitoring. Proceedings of the 5th International Conference on Pervasive Technologies Related to Assistive Environments. https://doi.org/10.1145/2413097.2413148

Twomey, N., Diethe, T., Fafoutis, X., Elsts, A., McConville, R., Flach, P., & Craddock, I. (2018). A comprehensive study of activity recognition using accelerometers. Information, 9(2), 27.

Additional Links

● GitHub URL: https://github.com/AAI530-Team6/final-team-project ● Tableau Public URL: https://public.tableau.com/app/profile/quang.tran2968/viz/ActivityPrediction/Dashboard1