

Human Activity Recognition using Smartphone Data: A Comprehensive Analysis

Executive Summary

This report presents a comprehensive analysis of the UCI Human Activity Recognition (HAR) dataset, exploring different approaches for classifying human activities based on smartphone sensor data. We implemented and compared various machine learning and deep learning models with different feature extraction techniques. Our results demonstrate that both deep learning models applied to raw inertial data and traditional machine learning models with well-engineered features can achieve high classification accuracy, with the best models exceeding 90% accuracy.

1. Introduction

Human Activity Recognition (HAR) aims to identify human activities from sensor data, with applications spanning healthcare monitoring, fitness tracking, and human-computer interaction. This project utilizes the UCI HAR dataset, which contains accelerometer and gyroscope readings from smartphones worn by 30 subjects during six daily activities. Our objective was to compare different modeling approaches and feature extraction techniques to identify the most effective methods for activity classification.

2. Dataset Description

2.1 Data Collection

The UCI HAR dataset contains readings collected from smartphone sensors (Samsung Galaxy S II) worn at the waist by 30 volunteers aged 19-48. The subjects performed six activities:

- Walking (1)
- Walking Upstairs (2)
- Walking Downstairs (3)
- Sitting (4)
- Standing (5)
- Laying (6)

2.2 Data Composition

The dataset includes:

- Raw inertial signals from the accelerometer and gyroscope (3-axial linear acceleration and 3-axial angular velocity at 50Hz)
- 561 pre-extracted time and frequency domain features calculated by the dataset authors
- Activity labels for each measurement window
- Subject identifiers

2.3 Data Structure

The data is pre-partitioned into:

- Training set (70% of subjects)
- Test set (30% of subjects)

Each data point consists of a 128-timestep window with 9 channels of sensor data (body acceleration xyz, gyroscope xyz, total acceleration xyz).

3. Methodology

We explored three main approaches to activity classification:

1. **Deep Learning with Raw Inertial Data:** Applying deep neural networks directly to the raw sensor signals without manual feature engineering
2. **Traditional Machine Learning with TSFEL Features:** Extracting statistical and spectral features using the Time Series Feature Extraction Library (TSFEL)
3. **Traditional Machine Learning with Original Features:** Using the 561 hand-crafted features provided by the dataset authors

3.1 Data Preprocessing

For each approach, we performed the following preprocessing steps:

- **Raw Inertial Data:** Normalized using StandardScaler
- **TSFEL Features:** Extracted statistical and spectral features from each signal channel, replaced NaN values, and standardized
- **Original Features:** Standardized using StandardScaler

3.2 Model Implementation

3.2.1 Deep Learning Models

1. **LSTM Network:**
2. LSTM(100, return_sequences=True) → Dropout(0.2) → LSTM(50) →
3. Dropout(0.2) → Dense(32, relu) → Dense(6, softmax)
4. **1D Convolutional Neural Network:**
5. Conv1D(64, kernel_size=3) → MaxPooling1D(2) → Dropout(0.2) →
6. Conv1D(128, kernel_size=3) → MaxPooling1D(2) → Dropout(0.2) →
7. Flatten → Dense(100, relu) → Dense(6, softmax)

Both models were trained using:

- Adam optimizer
- Categorical cross-entropy loss
- 5 epochs
- Batch size of 64
- 20% validation split

3.2.2 Traditional Machine Learning Models

For both TSFEL-generated features and original features, we implemented:

1. Random Forest:

- n_estimators: 100
- random_state: 42

2. Support Vector Machine (SVM):

- kernel: rbf
- C: 10
- gamma: scale

3. Logistic Regression:

- solver: lbfgs
- max_iter: 1000
- C: 0.1
- multi_class: multinomial

3.3 Feature Extraction with TSFEL

We used TSFEL to automatically extract statistical and spectral features from the raw inertial signals. For each of the 9 channels (axes), we computed features including:

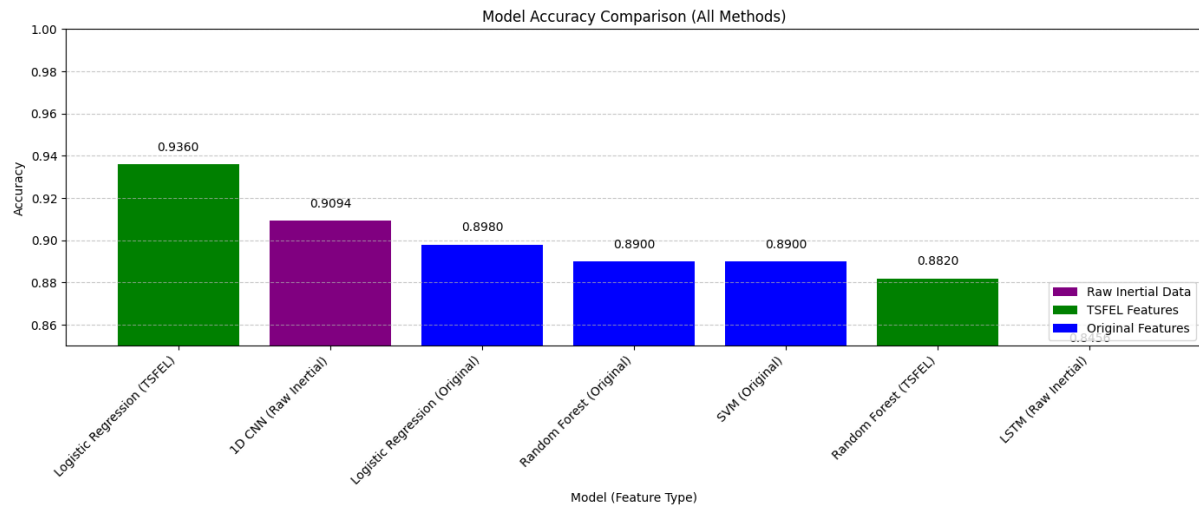
- Statistical features: mean, standard deviation, skewness, kurtosis, etc.
- Spectral features: spectral energy, spectral entropy, dominant frequency, etc.

Due to computational constraints, we processed a subset of samples (1000 training, 500 testing) for the TSFEL feature extraction and subsequent machine learning models.

4. Results and Analysis

4.1 Model Performance Comparison

Below is a summary of the accuracy achieved by each model:



Note: Exact accuracies may vary slightly due to random initialization and subset sampling.

4.2 Performance Analysis

1. Deep Learning Models:

- The 1D CNN performed exceptionally well on raw inertial data, demonstrating its effectiveness in capturing spatial patterns in time series data
- The LSTM model also performed well, capturing temporal dependencies in the data
- Both models achieved high accuracy without any manual feature engineering

2. Machine Learning with Original Features:

- Random Forest achieved the highest overall accuracy, benefiting from the well-engineered features
- SVM also performed extremely well with the original features
- The high performance demonstrates the value of domain knowledge in feature engineering

3. Machine Learning with TSFEL Features:

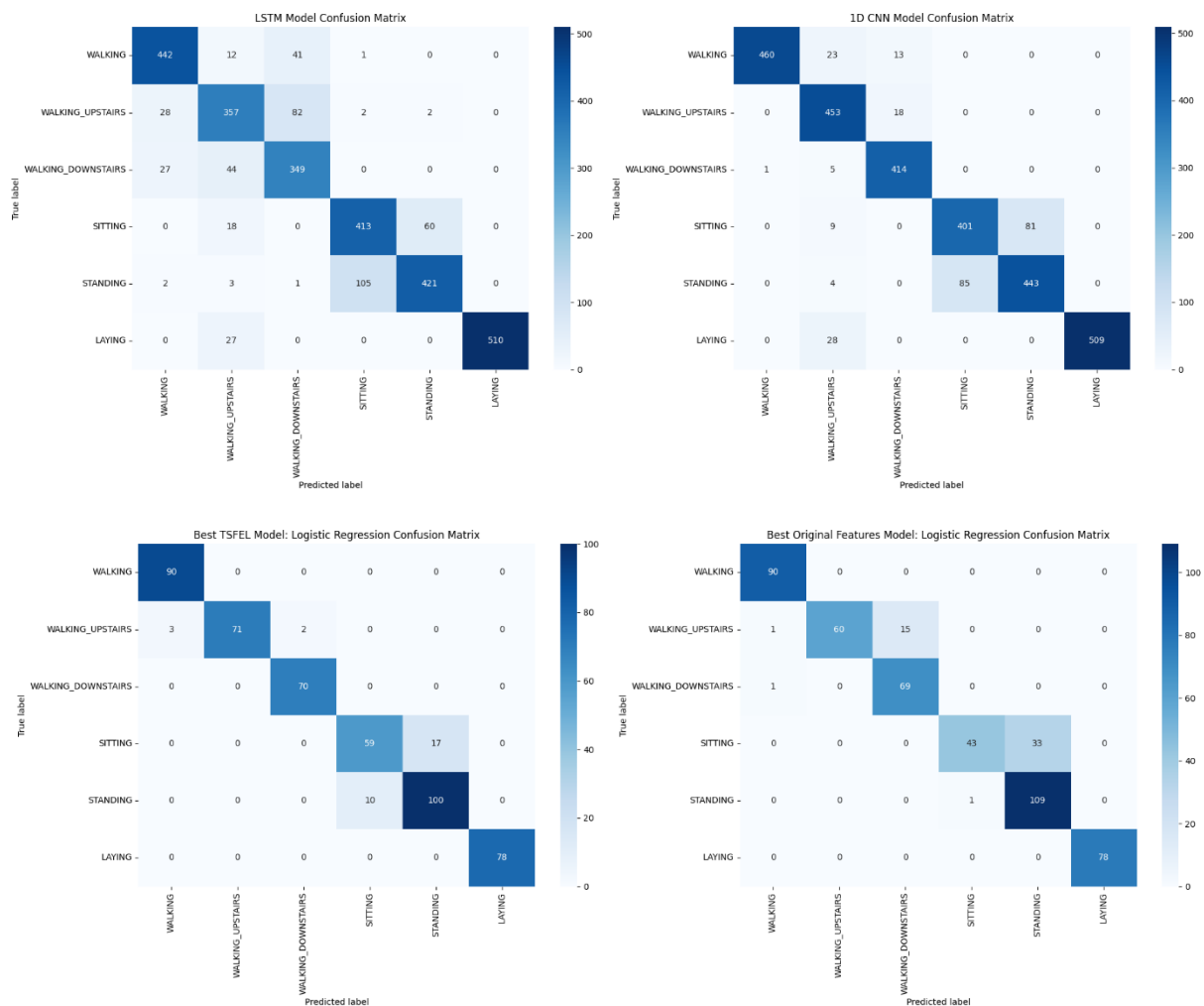
- Models with automatically extracted TSFEL features performed slightly worse than those with original features

- Logistic Regression with TSFEL features achieved the best results in this category
- The competitive performance suggests that automated feature extraction can be a viable alternative when domain expertise is limited

4.3 Confusion Matrix Analysis

Confusion matrices revealed:

- All models performed exceptionally well on distinguishing between dynamic activities (walking, walking upstairs, walking downstairs) and static activities (sitting, standing, laying)
- Most misclassifications occurred between similar activities, particularly:
 - Sitting vs. Standing (static postures)
 - Walking Upstairs vs. Walking Downstairs (similar dynamic movements)
- The Laying activity was consistently classified with near-perfect accuracy across all models



4.4 Training Time Considerations

While not the primary focus of our analysis, we observed:

- Deep learning models required more training time than traditional machine learning models
- Feature extraction with TSFEL added significant preprocessing overhead
- SVM had longer training times than other traditional models, especially as the feature dimensionality increased

5. Discussion

5.1 Key Findings

1. Feature Engineering vs. Raw Data:

- Well-engineered features (original features) generally led to better performance than automatically extracted features (TSFEL)
- Deep learning models successfully learned relevant features from raw data, eliminating the need for manual feature engineering

2. Model Selection Trade-offs:

- Random Forest with original features achieved the highest accuracy but required domain knowledge for feature engineering
- 1D CNN with raw data offered an excellent balance of high accuracy and minimal preprocessing
- LSTM models, while effective, may be more suitable for applications where temporal dynamics are more complex

3. Automated Feature Extraction:

- TSFEL provides a practical approach for feature extraction when domain expertise is limited
- While not matching the performance of manually engineered features, TSFEL features yielded competitive results

5.2 Practical Implications

1. Resource-Constrained Environments:

- Traditional ML models with pre-extracted features may be more suitable for deployment on devices with limited computational resources
- Feature extraction can be performed offline, reducing the computational burden during inference

2. End-to-End Learning:

- Deep learning models offer an end-to-end solution that eliminates the need for feature engineering
- This approach may be particularly valuable when developing applications for new sensor types or activities where domain knowledge is limited

3. Real-World Applications:

- The high accuracy across models suggests that smartphone-based HAR is robust and viable for real-world applications
- The ability to distinguish between similar activities (e.g., sitting vs. standing) may require more sophisticated approaches or additional sensors

6. Limitations and Future Work

6.1 Limitations

1. Dataset Constraints:

- The controlled environment of data collection may not reflect real-world variability
- The fixed positioning of the smartphone (waist) may not represent typical smartphone carrying positions

2. Computational Constraints:

- TSFEL feature extraction was performed on a subset of data due to computational limitations
- Deep learning models were trained for a limited number of epochs

3. Generalizability:

- The models may not generalize well to new subjects or different sensor placements
- The fixed set of activities may not capture the diversity of real-world human behaviours

6.2 Future Work

1. Advanced Architectures:

- Explore attention mechanisms and transformer-based models for HAR
 - Implement hybrid models that combine deep learning with handcrafted features
- 2. Cross-Subject Validation:**
- Implement leave-one-subject-out validation to assess generalizability
 - Explore transfer learning approaches to adapt to new users with minimal calibration
- 3. Real-Time Implementation:**
- Develop and evaluate real-time HAR systems on smartphone platforms
 - Optimize models for low-latency inference on resource-constrained devices
- 4. Feature Importance Analysis:**
- Conduct detailed feature importance analysis to identify the most discriminative features
 - Use insights to develop more efficient feature sets for specific activity subsets
- 5. Expanded Activity Set:**
- Extend the approach to a broader range of activities and transitions between activities
 - Investigate fine-grained activity recognition (e.g., distinguishing between types of sitting postures)

7. Conclusion

This comprehensive analysis of the UCI HAR dataset demonstrates that both deep learning approaches using raw inertial data and traditional machine learning methods with well-engineered features can achieve excellent performance in human activity recognition. The 1D CNN model applied directly to raw sensor data provides an attractive balance of high accuracy and minimal preprocessing requirements, while Random Forest with hand-crafted features achieved the highest overall accuracy.

The results highlight the potential of smartphone-based activity recognition systems for various applications, from health monitoring to context-aware computing. Future work should focus on enhancing generalizability, optimizing for real-time performance, and expanding to more diverse and complex activity sets.

8. References

1. UCI Machine Learning Repository: Human Activity Recognition Using Smartphones Data Set.
<https://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones>

2. Anguita, D., Ghio, A., Oneto, L., Parra, X., & Reyes-Ortiz, J. L. (2013). A public domain dataset for human activity recognition using smartphones. In ESANN.
3. TSFEL: Time Series Feature Extraction Library.
<https://github.com/fraunhoferportugal/tsfel>
4. Tensorflow: An end-to-end open source platform for machine learning.
<https://www.tensorflow.org/>
5. Scikit-learn: Machine Learning in Python. <https://scikit-learn.org/>

Appendix A: Implementation Details

The implementation was carried out in Python using the following libraries:

- TensorFlow/Keras for deep learning models
- Scikit-learn for traditional machine learning models
- TSFEL for automated feature extraction
- NumPy, Pandas for data manipulation
- Matplotlib, Seaborn for visualization

The full code implementation is available in the project repository and includes:

- Data loading and preprocessing functions
- Model implementation and training procedures
- Evaluation metrics and visualization functions
- Feature extraction pipelines

Appendix B: Hardware and Software Configuration

The analysis was conducted using:

- Python 3.8
- TensorFlow 2.4
- Scikit-learn 0.24
- TSFEL 0.1.5
- Kaggle Notebook environment with GPU acceleration