Human Activity Recognition Using Wi-Fi CSI Dataset for Room (1/1)

Introduction

The goal of Human Activity Recognition (HAR) is to use data gathered from several sensors to detect human actions including sitting, jogging, and walking. The gyroscopes and accelerometers found in wearable devices are the backbone of traditional HAR systems, however these sensors can be intrusive and inconvenient to use. A non-intrusive substitute is provided by Wi-Fi Channel State Information (CSI), which records the alterations in the wireless channel brought about by human motions. Without requiring users to carry any gadgets, we can properly identify various human actions by evaluating CSI data.

Objective

This document's objective is to provide a guide for implementing a Support Vector Machine (SVM) model to classify human activities based on a Wi-Fi CSI dataset. The document will cover data preprocessing, feature engineering, model training, evaluation, and interpretation of results.

Dataset Overview

The Wi-Fi CSI dataset used in this study consists of data captured by Wi-Fi devices that monitor changes in the channel state as people perform different activities in a controlled environment. Each data point in the dataset corresponds to a specific activity, and the CSI data is typically represented as a time series of complex numbers, reflecting amplitude and phase information.

Data Preprocessing

Preprocessing is crucial to prepare raw CSI data for training the SVM model. The steps include:

Normalization: Normalize the data to ensure that all features have a similar scale, which helps improve the performance of the SVM model.

Label Encoding: Activity labels were label encoded using label Enncoder to prepare them for categorical classification

Dataset Summary

Number of rows: 5228

Number of columns: 1026

Number of classes: 7

Support Vector Machine (SVM) Model

Support Vector Machine (SVM) is a supervised learning algorithm used for classification tasks. SVM finds the hyperplane that best separates the data into different classes by maximizing the margin between the nearest points (support vectors) of any class. The choice of kernel functions (linear, polynomial, Radial Basis Function) is crucial, as it determines the ability of the SVM to handle non-linear data.

Model Training

We split the dataset into training, validation, and test sets to evaluate the model's performance at different stages. Here's the process:

Training Set: 60% of the data, used to train the SVM model.

Validation Set: 20% of the data, used to validate the model during training and tune hyperparameters.

Test Set: 20% of the data, used for final evaluation after training is complete.

Later we begin by initializing the SVM model with a linear kernel. The SVM tries to find the optimal hyperplane that best separates the activity classes. To simulate training over multiple epochs, we iteratively train the model and evaluate its accuracy on both the training and validation sets. The goal during training is to observe whether the model's accuracy improves over time without overfitting on the training data.

Hyperparameter tuning

Hyperparameters are parameters that control the learning process of the model and are not learned from the data itself. For SVM, the most important hyperparameters are:

C (Regularization Parameter): Controls the trade-off between classifying training points correctly and maintaining a large margin.

Kernel Type: Defines how the SVM constructs the decision boundary (e.g., linear, RBF, polynomial).

Gamma: Controls how far the influence of a single training point reaches (applies to non-linear kernels like RBF, polynomial, and sigmoid).

Degree: Used for polynomial kernel to control the complexity of the decision boundary.

Results

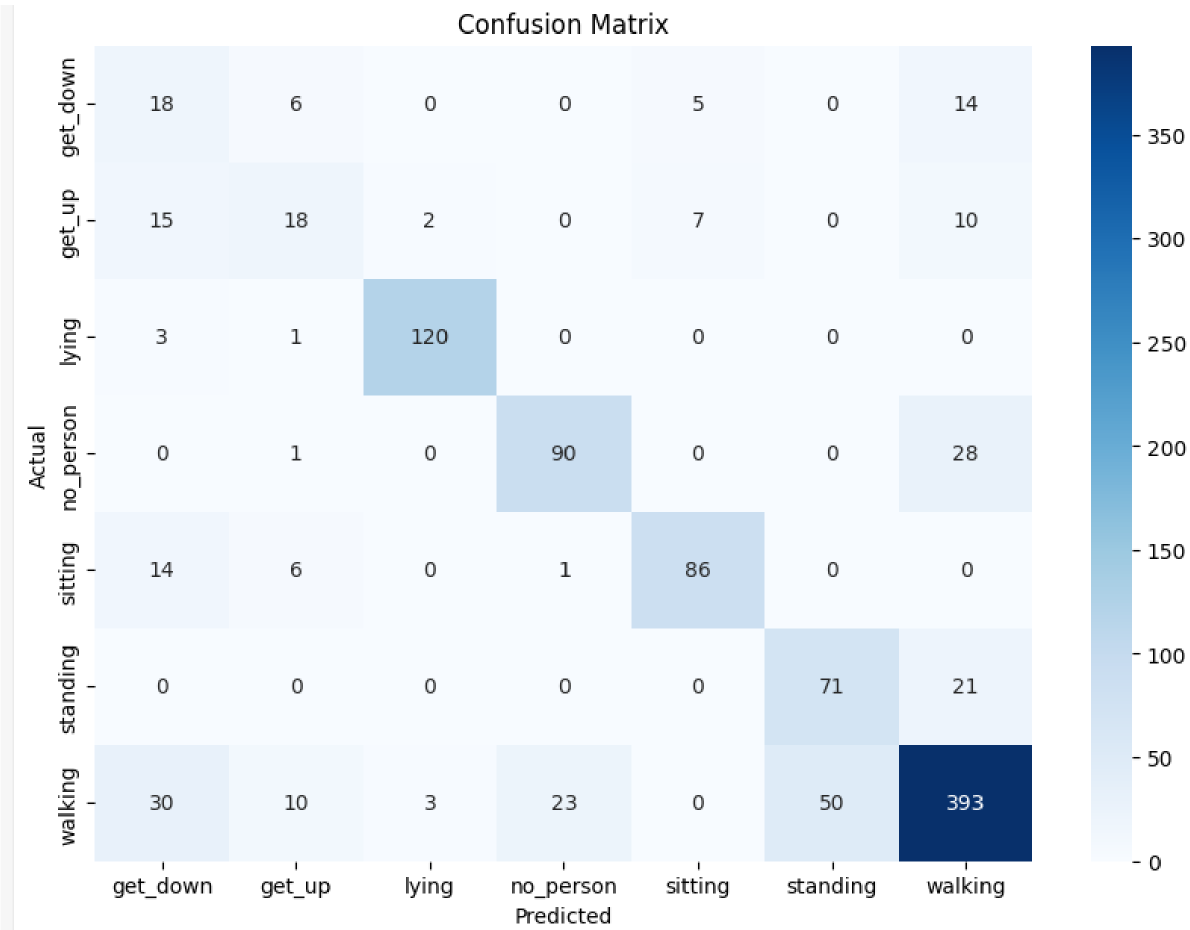
The final test accuracy was reported to be 76% and the classification report is given below. As well as **Precision, Recall, F1-Score were defined where** the F1-scores for most activities were high, suggesting the model balances precision and recall effectively across different classes.

```
Final Test Accuracy: 0.76

Classification Report:
              precision    recall  f1-score   support

    get_down       0.23      0.42      0.29        43
      get_up       0.43      0.35      0.38        52
       lying       0.96      0.97      0.96       124
   no_person       0.79      0.76      0.77       119
     sitting       0.88      0.80      0.84       107
    standing       0.59      0.77      0.67        92
     walking       0.84      0.77      0.81       509

    accuracy                           0.76      1046
   macro avg       0.67      0.69      0.67      1046
weighted avg       0.79      0.76      0.77      1046
```



After hyperparameter tuning Best Parameters identified were 'C': 10, 'degree': 2, 'gamma': 'auto', 'kernel': 'rbf'.

The accuracy achieved was 84%

```
Test Set Classification Report:
              precision    recall  f1-score   support

    get_down       0.34      0.33      0.33        43
      get_up       0.50      0.40      0.45        52
       lying       0.96      0.97      0.96       124
   no_person       0.89      0.84      0.87       119
     sitting       0.89      0.87      0.88       107
    standing       0.74      0.79      0.76        92
     walking       0.88      0.90      0.89       509

    accuracy                           0.84      1046
   macro avg       0.74      0.73      0.73      1046
weighted avg       0.84      0.84      0.84      1046
```

Future Work:

Use of Deep Learning Models

While the SVM model performed well, further improvements could be made by applying deep learning techniques such as Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs). These models could better capture temporal patterns and spatial relationships in CSI data.

Data Augmentation

Expanding the dataset by augmenting the CSI data (e.g., adding noise or simulating different environments) could improve model robustness and help generalize to new settings.

Real-time HAR Implementation

Future work could focus on implementing the HAR model in real-time, allowing it to detect activities in dynamic environments.

Additional Feature Engineering

Investigating other feature extraction techniques (e.g., wavelet transforms or complex-valued features) could provide more insights into the underlying patterns in CSI data.

Conclusion

This document presented a detailed implementation of an SVM model for Wi-Fi CSI data-based Human Activity Recognition in this article. Preprocessed data was used to train the model, and in order to maximize performance, we investigated hyperparameter optimization. The outcomes showed how well the SVM model performs when it comes to categorizing various human activities according to wireless signal data.

We were able to observe that the model worked well on unknown data by using confusion matrices and classification reports. It achieved high accuracy and good precision-recall metrics across activity classes. Though misclassified activities can still be improved, future research can solve this by utilizing cutting-edge methods like deep learning or better feature engineering.

This study lays the groundwork for more sophisticated models and real-time implementations while showcasing the potential of Wi-Fi CSI for non-invasive Human Activity Recognition.