CS 6762

SP, ML, FC

Programming Assignment 2 Report

Jinyoon Kim(xna8aw)**, Yusen Wu**(xrn4ub)**, Katz Yan**(dda7pk)

Disclaimer: The results presented in this report are based on data collected by the team members and processed on our specific development environments. Due to variations in data collection, Java/Weka library versions, and underlying hardware, results may vary slightly when run in a different environment.

This report details the implementation and results for Programming Assignment 2. The objective of this assignment was to build a programmatic Java pipeline to improve upon the "Hand Washing" gesture recognition model from Assignment 1. This was achieved by systematically adding more data, tuning parameters like time-slice windows, engineering new features, performing feature selection, and comparing multiple machine learning classifiers.

Part 1: Baseline Accuracy with Combined Data

Assignment: This part required collecting additional data (for both "hand_wash" and "non_hand_wash" classes) and combining it with the dataset from Assignment 1. A Decision Tree (J48) classifier was then programmatically trained on this new, larger dataset using the original 1-second window and 6 features (mean/std for each axis) to establish a new baseline.

Results:

*(Please add your results here. Be sure to include:)

- Accuracy from Assignment 1: 97.90% (on 3,529 instances)
- Accuracy from Part 1 (Combined Data): 96.48% (on 6,940 instances)

The new baseline was established using a combined dataset of 6,940 instances. The Decision Tree (J48) classifier, evaluated with 10-fold cross-validation, achieved an accuracy of **96.48%**.

The confusion matrix for this new baseline is as follows:

	Predicted: hand_wash	**Predicted: non_hand_wash**

Actual: hand_wash	3745	93
Actual: non_hand_wash	151	2951

Did the accuracy improve?

No, the accuracy slightly decreased from 97.90% to 96.48%.

Analysis:

This small decrease in accuracy, despite nearly doubling the number of instances (from 3,529 to 6,940), is a significant finding. It suggests that the new data collected for Assignment 2 introduced more complexity or ambiguity to the dataset. For example, the new "non_hand_wash" activities may have included gestures that are more similar to hand washing than the original data.

While the model is now trained on a much larger and more robust dataset, that dataset is also inherently "harder" to classify, resulting in a minor drop in overall accuracy. The model is less "overfit" to a smaller, simpler dataset and is now more representative of a real-world scenario.

Part 2: Time Slice Window Analysis

Assignment: This part involved exploring the impact of the time window size on classification accuracy. Using the combined dataset from Part 1, the Decision Tree classifier was re-run using window sizes of 2, 3, and 4 seconds, all with a 1-second sliding window. The original 6 features (mean/std) were used.

Results:

Window Size (1s slide)	**Accuracy**	**Status**
1 second (from Part 1)	96.73%	
2 seconds	97.67%	

3 seconds	98.14%	
4 seconds	98.63%	**Best**

Analysis: The optimal time slice was the **4.0-second** window, which achieved an accuracy of **98.63%**. The results show a clear trend: as the window size increased from 1 to 4 seconds, the accuracy consistently improved. This suggests that a 4-second window captures a more complete and distinct pattern for "hand washing" than a shorter window, allowing the classifier to make more accurate decisions.

Answer: Did the accuracy improve compared to the 1-second window?

Yes, the accuracy improved significantly, rising from **96.73%** at **1 second** to **98.63%** at **4 seconds**. The 4.0-second window will be used for all subsequent parts of the assignment.

Part 3: Expanded Feature Set Analysis

Assignment: This part expanded the feature set from 6 to **12 features**. Using the best-performing time interval identified in Part 2 (4.0 seconds), **four features (mean, standard deviation, median, and Root Mean Square (RMS))** were computed for each of the three axes. The Decision Tree classifier was then retrained on this new 12-feature dataset.

Results:

- **Best Accuracy from Part 2** (4-second window, 6 features): **98.63%**
- **Accuracy from Part 3** (4-second window, 12 features): **98.62%**

The confusion matrix for the 12-feature model is as follows:

	Predicted: hand_wash	**Predicted: non_hand_wash**
Actual: hand_wash	3324	37

Actual: non_hand_wash	52	3015

Answer: Did the accuracy improve compared to Part 2?

No, the accuracy did not improve. It saw a negligible decrease from 98.63% to 98.62%.

Analysis:

Adding the **median** and **RMS** features did not provide any meaningful improvement to the model's accuracy. This indicates that the original 6 features (mean and std) already captured nearly all of the useful variance needed to distinguish between the two classes, at least for the Decision Tree classifier. The new features were likely redundant or added a tiny amount of noise, leading to a statistically insignificant change in performance. This is a valuable finding, as it suggests that a simpler 6-feature model is just as effective as the more complex 12-feature model.

Part 4: Sequential Feature Selection (SFS) with Decision Tree

Assignment: This part involved performing automated feature selection. Using the best time interval from Part 2 and the full 12-feature set from Part 3, a Sequential Feature Selection (SFS) algorithm (e.g., BestFirst) was run using the Decision Tree classifier as the evaluator.

Results:

Selected Features: F9, F1, F0, F5, F11

Accuracy Comparison:

Accuracy with all 12 features (from Part 3): 98.62%

Accuracy with SFS-selected features: 99.12%

Accuracy Progression: (Optional but recommended) Show how accuracy changed as SFS added features one by one (if your tool provides this).

Added feature F9 → accuracy = 91.024%

Added feature F1 → accuracy = 95.769%

Added feature F0 → accuracy = 98.102%

Added feature F5 → accuracy = 98.740%

Added feature F11 → accuracy = 98.83%

Answer: Which features were selected? Does accuracy improve compared to Part Three?

The features F9, F1, F0, F5, F11 are selected, which correspond to std_z, std_x, mean_x, std_y, rms_z. The accuracy improves from 98.62% to 98.83%

Part 5: SFS with Random Forest and SVM & Final Comparison

This final part repeated the SFS process from Part 4 two more times, using Random Forest and Support Vector Machine (SVM) as the classifiers. The goal was to compare the feature subsets and final accuracy of all three models (Decision Tree, Random Forest, and SVM).

Results:

Random Forest:

- Selected Features (SFS + RF): [9, 1, 0, 5, 7, 8]
- Accuracy Progression:

Iteration	Added Feature	Accuracy (%)
1	F9	87.01
2	F1	95.88
3	F0	98.41
4	F5	99.07
5	F7	99.36
6	F8	99.38

- Final Accuracy (SFS + RF): 99.38%

Support Vector Machine (SVM):

- Selected Features (SFS + SVM): [9, 1, 5, 10, 6, 2, 11, 4, 7, 0]
- Accuracy Progression:

Iteration	Added Feature	Accuracy (%)
1	F9	90.76
2	F1	95.72
3	F5	96.89
4	F10	97.62
5	F6	97.76
6	F2	98.02
7	F11	98.41
8	F4	98.49
9	F7	98.54
10	F0	98.58

- Final Accuracy (SFS + SVM): 98.58%

Final Comparison:

Classifier	**Feature Set**	**Final Accuracy**
Decision Tree	[9, 0, 1, 5, 4]	98.77%
Random Forest	[9, 1, 0, 5, 7, 8]	99.38%
SVM	[9, 1, 5, 10, 6, 2, 11, 4, 7, 0]	98.58%

Answer: Out of the three classifiers, which was the best?

Among the three classifiers, **Random Forest** performed the best, achieving the highest final accuracy of 99.38% with only six selected features.

Its ensemble structure allowed it to capture diverse patterns and generalize better than the other models.

The Decision Tree achieved a slightly lower accuracy of 98.77% using five features, showing good performance with a simple model, while the SVM reached 98.58% but required ten features.

Overall, Random Forest offered the best balance between accuracy and feature efficiency.