Protocol Weka

Prerequisite

- Weka Version: 3.8.0
- Interruption Data Set with Normalized Features

Feature Selection

Open Weka "Explorer"

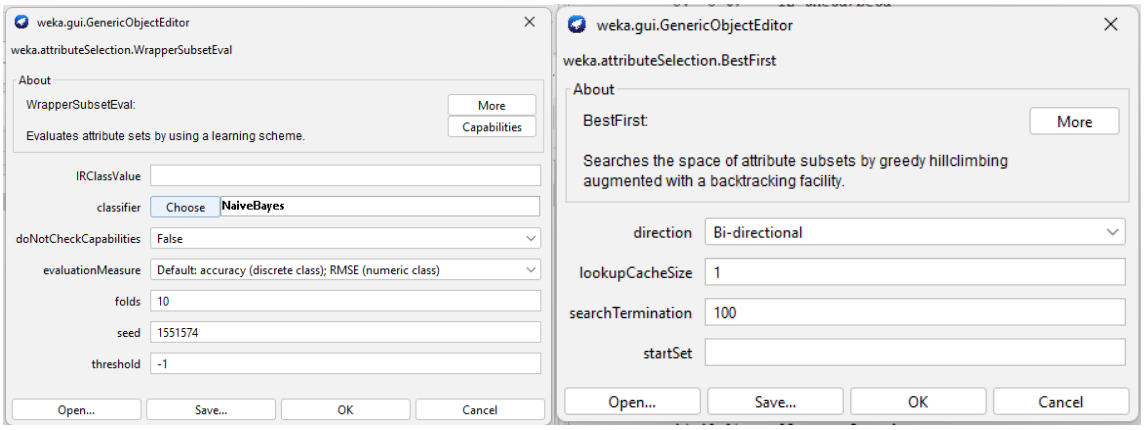
Feature Selection: 2-State

1. I used CfsEvalSubset with Cross-Validation.
2. This led to unexpected behaviour: Only a single feature was selected.
3. I reran CfsEvalSubset on the whole Data Set to be able to inspect the Merit Score
    a. This led to 0 Merit for the selected features.
    b. Interpretation: CfsEvalSubset does not detect any "relatedness" between dependent and independent variables

```
=== Attribute Selection on all input data ===

Search Method:
        Best first.
        Start set: no attributes
        Search direction: bi-directional
        Stale search after 100 node expansions
        Total number of subsets evaluated: 3791
        Merit of best subset found:    0

Attribute Subset Evaluator (supervised, Class (nominal): 39 interruptible):
        CFS Subset Evaluator
        Including locally predictive attributes

Selected attributes: 1 : 1
                     delta
```

4. Switching to Wrapper-based Feature Selector to reduce feature set for better comparability:

Initial Assumption: 10s Time Windows with NaiveBayes achieve best results (Time Windows will be tested later on).



Selected Features:

1. delta
2. theta/gamma
3. alpha/gamma
4. num_of_peaks

5. mean_temp

Feature Selection: 5-State

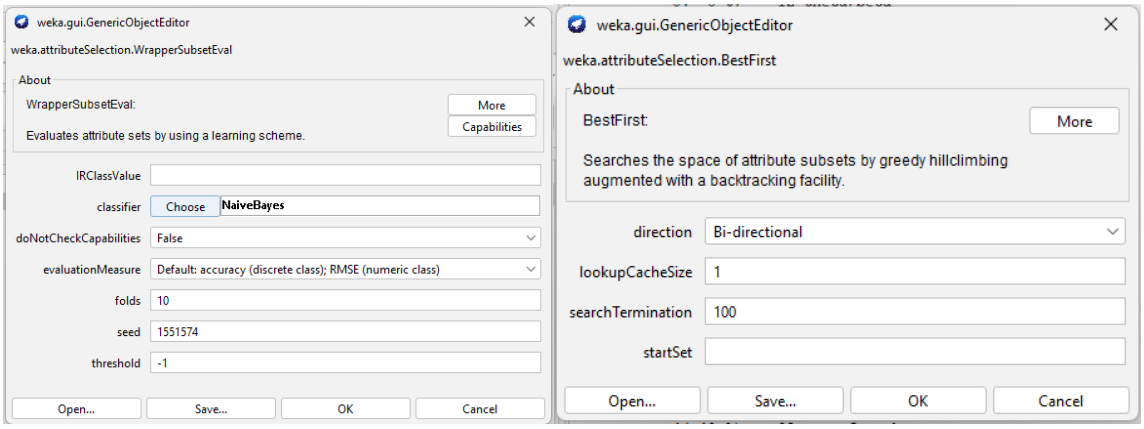
Repeat Steps from 2-State Classification

1. I used CfsEvalSubset with Cross-Validation.
2. This led to unexpected behaviour: Only a single feature was selected.
3. I reran CfsEvalSubset on the whole Data Set to be able to inspect the Merit Score
    a. This led to 0.173 Merit for the selected features.
    b. Interpretation: CfsEvalSubset does not detect close to no "relatedness" between dependent and independent variables

```
=== Attribute Selection on all input data ===

Search Method:
        Greedy Stepwise (forwards).
        Start set: no attributes
        Merit of best subset found:    0.173

Attribute Subset Evaluator (supervised, Class (nominal): 39  interruptibility):
        CFS Subset Evaluator
        Including locally predictive attributes

Selected attributes: 13 : 1
                     theta/gamma
```

4. Switching to Wrapper-based Feature Selector to reduce feature set for better comparability:

Initial Assumption: 10s Time Windows with NaiveBayes achieve best results (Time Windows will be tested later on).



Feature Selection:

1. alpha/gamma
2. max_peak_amplitude
3. pnn50

Building Classifier

Initial Test of Models using Simple 10-Fold Cross Validation.

2-State

Performance Majority Classifier (ZeroR): 60.9756 %

Performance Naive Bayes: 74,3902% (10-k CV / Seed = Matriculation Number:1551574)

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          61               74.3902 %
Incorrectly Classified Instances        21               25.6098 %
Kappa statistic                          0.4398
Mean absolute error                      0.3286
Root mean squared error                  0.4631
Relative absolute error                 68.9414 %
Root relative squared error             94.8937 %
Total Number of Instances               82

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0,563     0,140     0,720       0,563   0,632      0,448    0,718     0,592     True
               0,860     0,438     0,754       0,860   0,804      0,448    0,718     0,775     False
Weighted Avg.  0,744     0,321     0,741       0,744   0,737      0,448    0,718     0,704

=== Confusion Matrix ===

  a  b   <-- classified as
 18 14 |  a = True
  7 43 |  b = False
```

5-State

Performance Majority Classifier (ZeroR): 35.3659 %

Performance Naive Bayes: 46.3415 % (10-k CV / Seed = Matriculation Number:1551574)

```
Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          38               46.3415 %
Incorrectly Classified Instances        44               53.6585 %
Kappa statistic                          0.2685
Mean absolute error                      0.2355
Root mean squared error                  0.4036
Relative absolute error                 77.2951 %
Root relative squared error            103.5215 %
Total Number of Instances               82

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0,545     0,085     0,500       0,545   0,522      0,444    0,703     0,361     1
               0,690     0,396     0,488       0,690   0,571      0,281    0,682     0,542     2
               0,400     0,056     0,500       0,400   0,444      0,380    0,794     0,460     3
               0,238     0,180     0,313       0,238   0,270      0,064    0,585     0,329     4
               0,273     0,028     0,600       0,273   0,375      0,348    0,572     0,444     5
Weighted Avg.  0,463     0,208     0,461       0,463   0,446      0,268    0,659     0,440

=== Confusion Matrix ===

  a  b  c  d  e   <-- classified as
  6  3  1  1  0 |  a = 1
  1 20  1  5  2 |  b = 2
  0  3  4  3  0 |  c = 3
  3 12  1  5  0 |  d = 4
  2  3  1  2  3 |  e = 5
```

Manually Creating Leave One Subject Out Folds

pID: Needs to be converted from Numeric to Nominal first! (ID:39 in the Screenshots points at Column with pID. (Note: The column ID changes after feature selection.)

Train Set: All participants except one. Test Set: Left out participant.

Repeat for 10 participants.

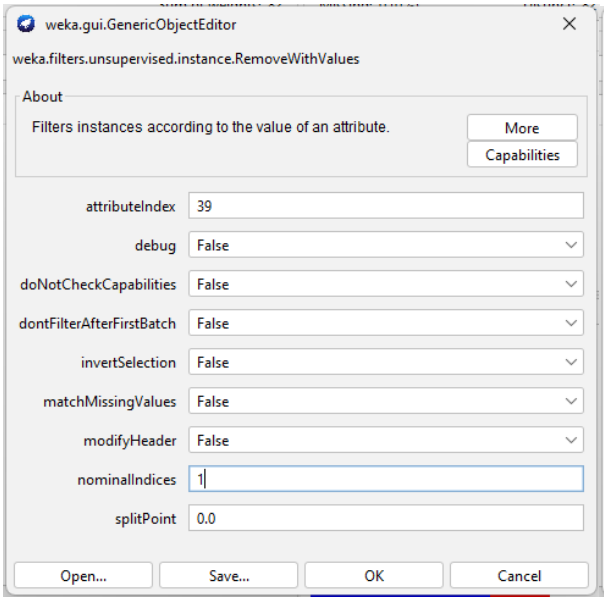


Figure 1 Filtering for Test Set

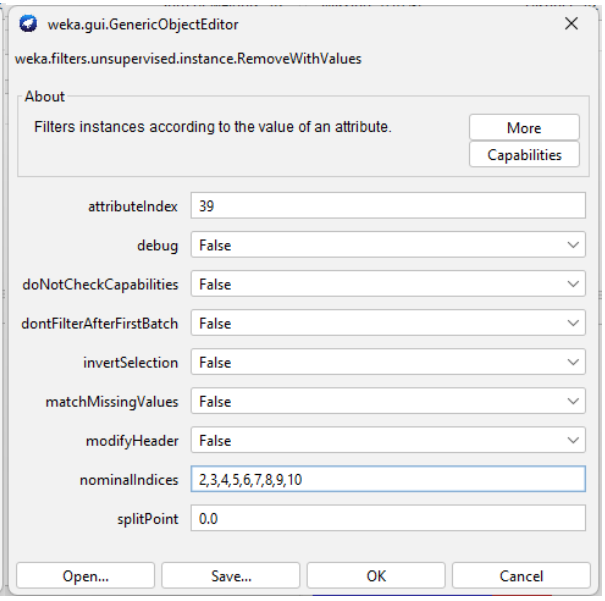


Figure 2 Filtering For Train Set

Remove pID Column.

Saving 10 Training Sets and 10 Test Sets.

Repeat for 5 State Data Set.

Validation

Comparing Classification Results

For validation of the classifier performance, it is unclear what the base performance is. From the original paper it seems as if they took the performance of a majority classifier being used once as the baseline.

Therefore, I assume they did a one-sided T-test when testing for a significant difference in performance between the baseline and the accuracy samples of the cross-validations.

Another possible approach would have been to build a majority classifier for each cross-validation fold as a baseline model and do a (corrected) two-sided T-Test instead.

Repeated Cross Validation

10 times 10-Fold CV

Validating the Naïve Bayes Classifier against a Majority Vote Classifier (ZeroR)

Source Code:

- PerInstanceValidation2State.java
- PerInstanceValidation5State.java

10 times 10-Fold CV Results: 2-State

10 times 10-Fold CV Results: 5-State

Per Participant Cross Validation

(Aka Leave One Subject Out)

Source Code:

- PerParticipantValidation2State.java
- PerParticipantValidation5State.java

Summary of Cross Validation Results

(*One Sample T-Test P-Value < 0.05)

States	Per Instance CV			Per Participant CV			Majority Classifier
	Accuracy	*Cohen's Kappa*	*Stdev*	*Accuracy*	*Cohen's Kappa*	*Stdev*	
Two	76%*	0.48	13.1	71%	0.11	16.4	60.98%
Five	46*	0.26	14.8	48%	0.22	21.6	35.37%

Results of the Original Study:

# States	Study	Per Instance CV			Per Participant CV			Majority Classifier
		Accuracy	Cohen's Kappa	Stdev	Accuracy	Cohen's Kappa	Stdev	Accuracy
Two	Lab	91.5%*	0.65	0.7	74.9%	-0.11	36.0	83.3%
	Field	78.6%*	0.44	1.3	69.4%	0.22	19.7	70.7%
Five	Lab	43.9%*	0.18	2.9	37.6%	0.11	21.7	38.9%
	Field	32.5%	0.13	2.1	28.2%	0.07	14.2	32.4%

Table 2.1: Classification results by number of states and study, for per instance and per participant cross-validation (CV), compared to a majority classifier as a baseline value (* indicates that there is a significant difference in accuracy to the majority classifier).

Comparing the results of the original study with my own, although there is a comparable improvement on baseline accuracy, in my validation the accuracy varied far stronger in the repeated cross validation (Stdev[2-State]: 13,5 vs 0,7; Stdev[5-State]: 16,3 vs 2,9) This means that the final performance of our model may be only a bit better (or worse) than a simple majority classifier.

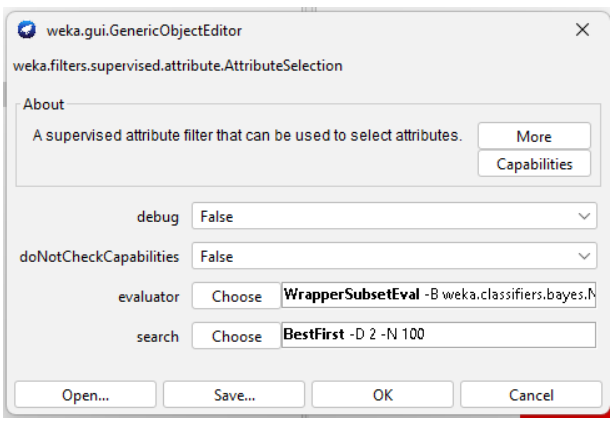
Comparing Time Windows

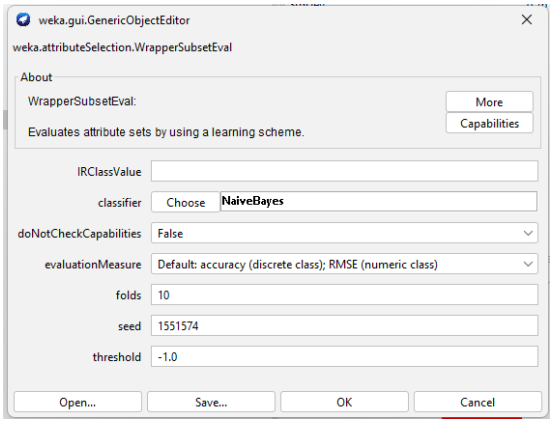
Again, I will need to select the time windows based on wrapper-based attribute selectors, since CFSEval does not work on my Data Set.

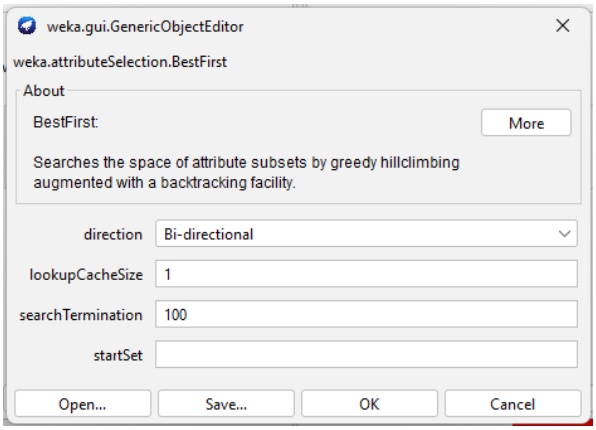
I assume they validated all time windows using 10 times 10-fold Cross Validation since the points for the 10 seconds time window on their comparison graphic match the values in the table from the repeated cross validation.

Wrapper Attribute Selection Protocol

Apply Wrapper-base Feature Selection to each Time Window Data Set.







Save filtered Data Set as:

"[Project Path]\Normalized Data\[Time Window]\with_pID\normalized_interruption_data_ALL_tw_[Time Window]s_pID_3hz_[2state/5state]_filtered_attributes.arff"

Attribute Selection (TWs)

Attribute Selection 10s 2-State:

CFSEval (Train Set): (Merit: 0)

Wrapper:

6. delta
7. theta/gamma
8. alpha/gamma
9. num_of_peaks
10. mean_temp

Attribute Selection 20s 2-State:

CFSEval (Train Set): (Merit: 0,272)

1. alpha/gamma
2. max_peak_amplitude
3. sum_peak_amplitude

Wrapper:

1. theta
2. delta/theta

3. delta/gamma
4. theta/alpha
5. theta/gamma
6. num_of_peaks
7. mean_temp

Attribute Selection 30s 2-State:

CFSEval (Train Set): (Merit: 0)

Wrapper:

1. alpha
2. theta/gamma
3. mean_hr_in_bpm
4. num_of_peaks
5. mean_temp

Attribute Selection 45s 2-State:

CFSEval (Train Set): (Merit: 0)

Wrapper:

1. theta
2. alpha
3. delta/gamma
4. mean_hr_in_bpm
5. num_of_peaks
6. mean_temp

Attribute Selection 60s 2-State:

CFSEval (Train Set): (Merit: 0)

Wrapper:

1. beta
2. delta/alpha
3. delta/gamma
4. mean_hr_in_bpm
5. sdnn
6. pnn50
7. max_temp
8. num_of_peaks
9. mean_temp

Attribute Selection 120s 2-State:

CFSEval (Train Set): (Merit: 0)

Wrapper:

1. delta/alpha
2. theta/beta
3. gamma/alpha
4. num_of_blinks
5. hr_variance_in_bpm

6. max_temp

Attribute Selection 10s 5-State:

CFSEval (Train Set): (Merit: 0.173)

1. theta/gamma

Wrapper:

4. alpha/gamma
5. max_peak_amplitude
6. pnn50

Attribute Selection 20s 5-State:

CFSEval (Train Set): (Merit: 0)

Wrapper:

1. delta/alpha
2. theta/delta
3. alpha/gamma
4. hr_variance_in_bpm
5. max_peak_amplitude
6. sum_peak_amplitude
7. pnn50

Attribute Selection 30s 5-State:

CFSEval (Train Set): (Merit: 0)

Wrapper:

1. theta/delta
2. hr_variance_in_bpm

Attribute Selection 45s 5-State:

CFSEval (Train Set): (Merit: 0)

Wrapper:

1. delta/theta
2. gamma/beta
3. num_of_peaks

Attribute Selection 60s 5-State:

CFSEval (Train Set): (Merit: 0)

Wrapper:

1. beta/gamma
2. gamma/beta
3. num_of_peaks
4. sum_peak_amplitude
5. pnn20
6. mean_temp

Attribute Selection 120s 5-State:

CFSEval (Train Set): (Merit: 0)

Wrapper:

1. beta/gamma
2. gamma/beta
3. pnn50

Attribute Selection 180s 5-State:

CFSEval (Train Set): (Merit: 0)

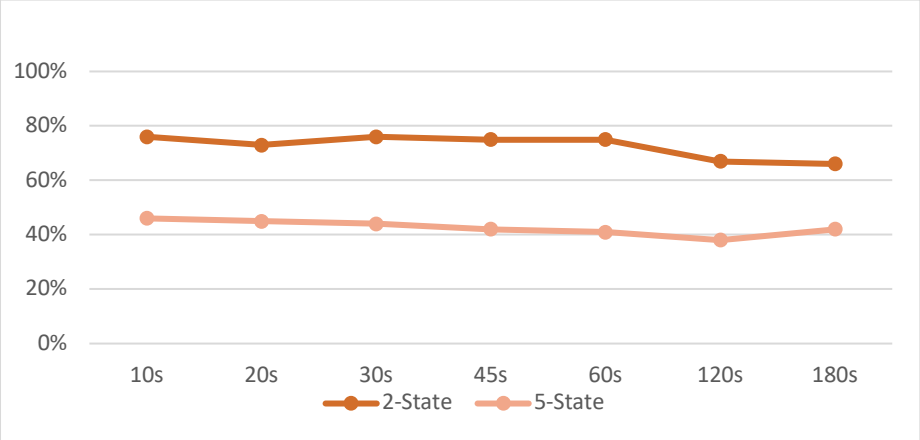
Wrapper:

1. theta/alpha
2. alpha/delta
3. gamma/beta
4. hr_variance_in_bpm
5. max_temp

Results: Comparing Time Windows

As observed in the original study shorter time windows (<60s) perform better than longer ones. I would also have the same intuition as the original author, that the shorter time windows probably benefit from having less artifacts. As in the original study this trend is much weaker for the 5-State predictions.

	2-State	**5-State**
10s	76%	46%
20s	73%	45%
30s	76%	44%
45s	75%	42%
60s	75%	41%
120s	67%	38%
180s	66%	42%



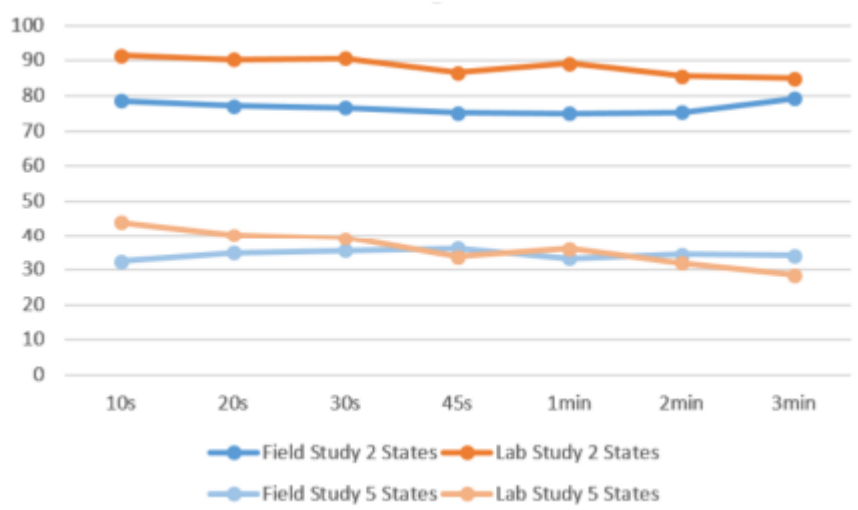


Figure 3 Graph from Original Paper