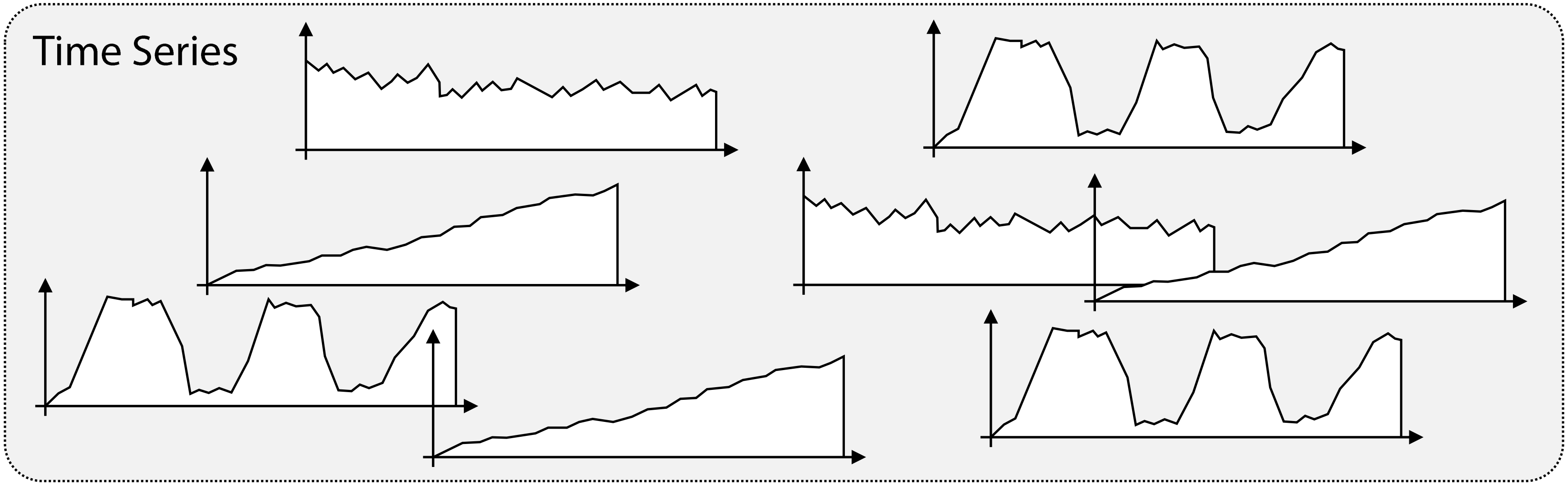
Selecting Time Series Clustering Methods based on Run-Time Costs

Andreas Schörgenhumer
Paul Grünbacher
Hanspeter Mössenböck

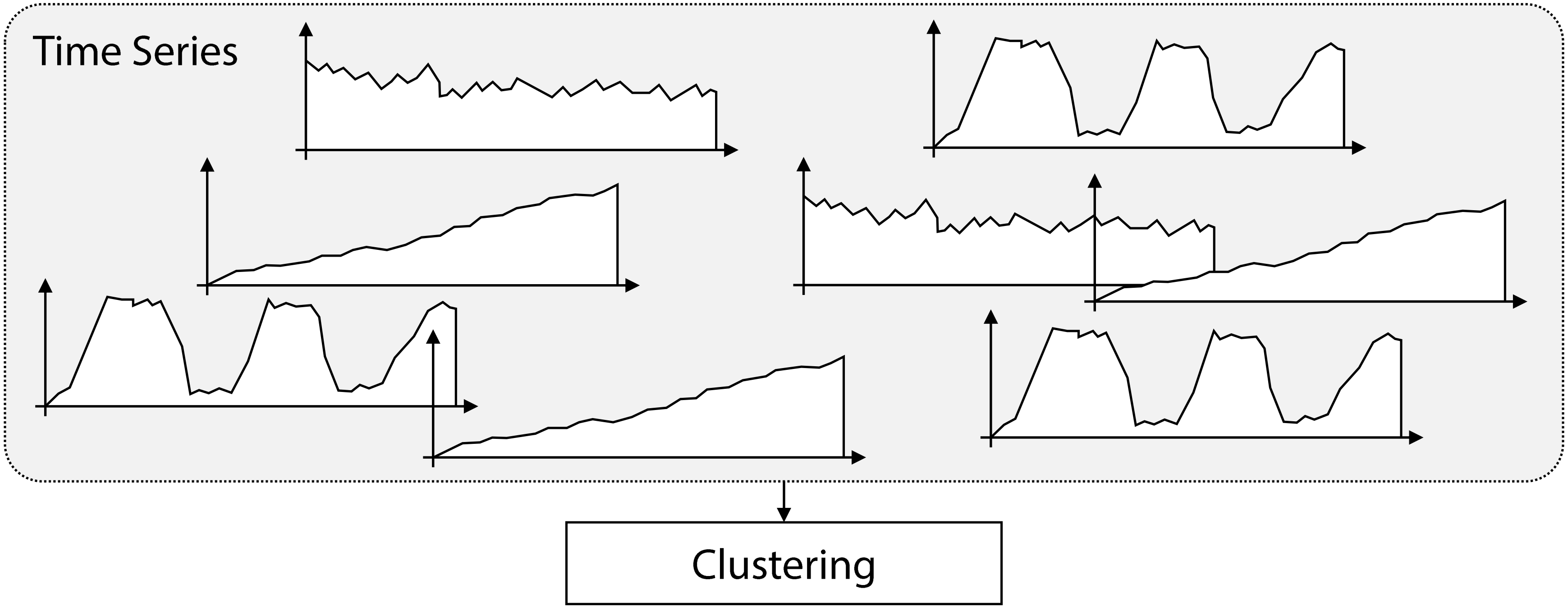
12.11.2020

JOHANNES KEPLER UNIVERSITY LINZ

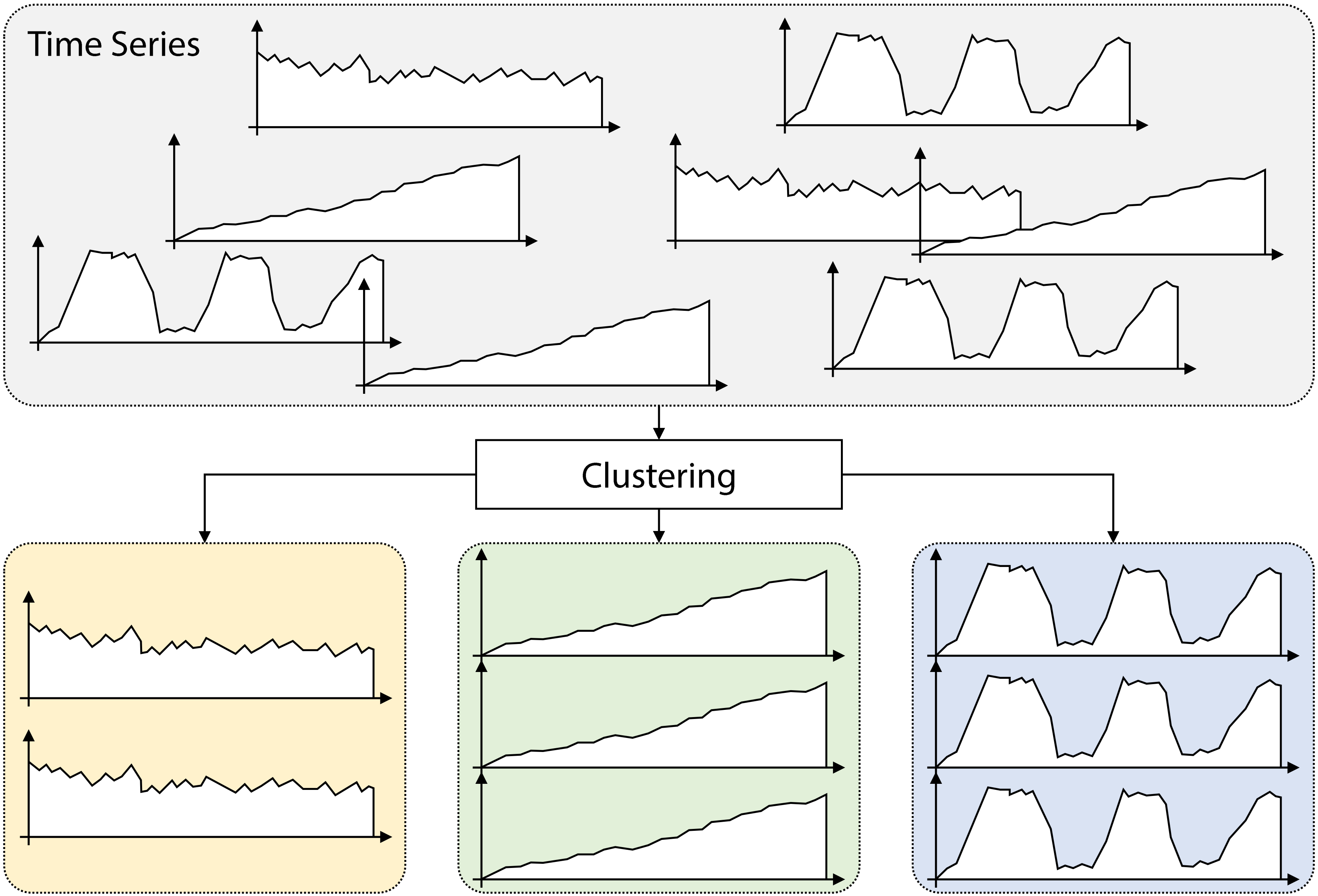
Motivation



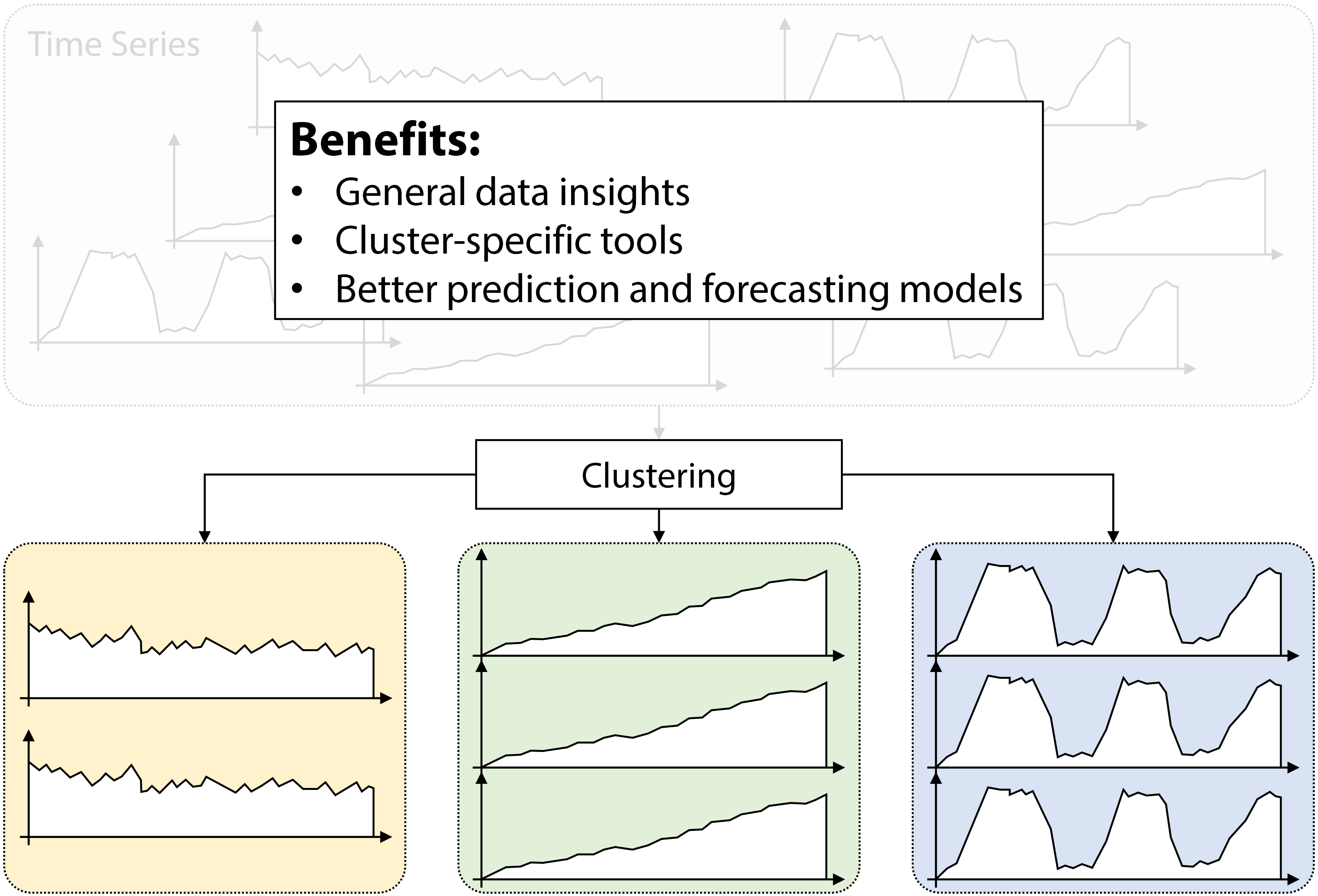
Motivation



Motivation

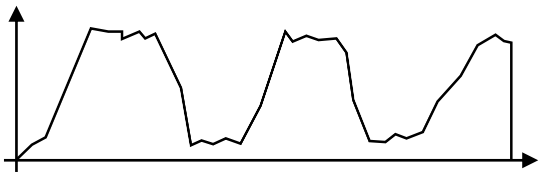


Motivation


Time Series
Benefits:
• General data insights
• Cluster-specific tools
• Better prediction and forecasting models
Clustering

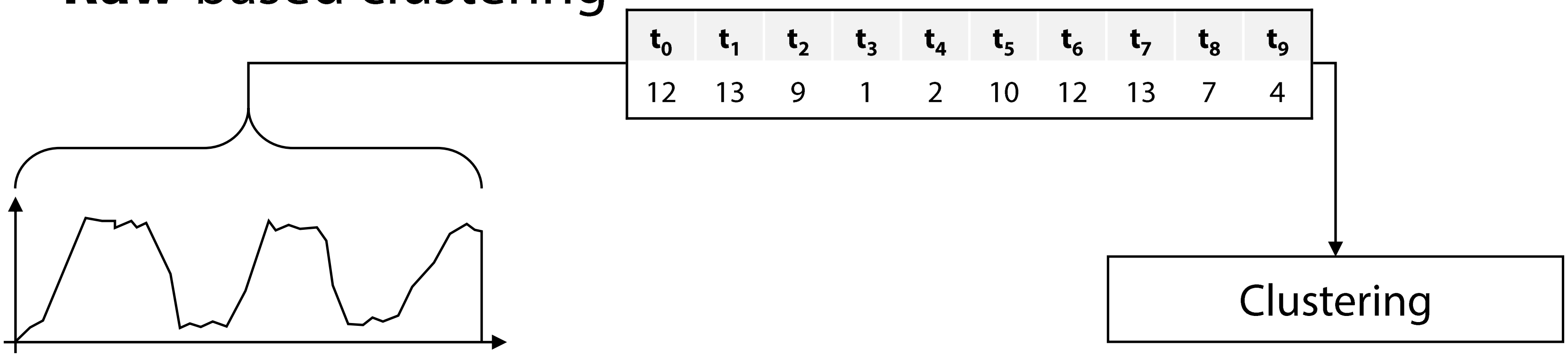
Clustering

- **Raw**-based clustering



Clustering

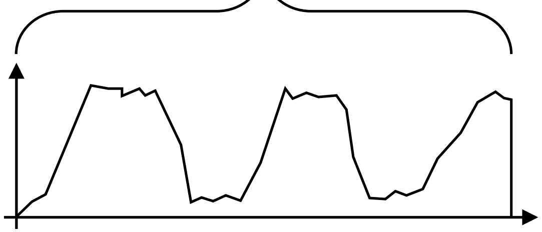
- **Raw**-based clustering



t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9
12	13	9	1	2	10	12	13	7	4

Clustering

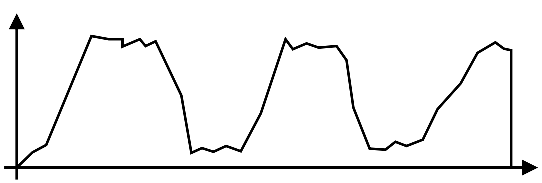
- **Raw**-based clustering



t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9
12	13	9	1	2	10	12	13	7	4

Clustering

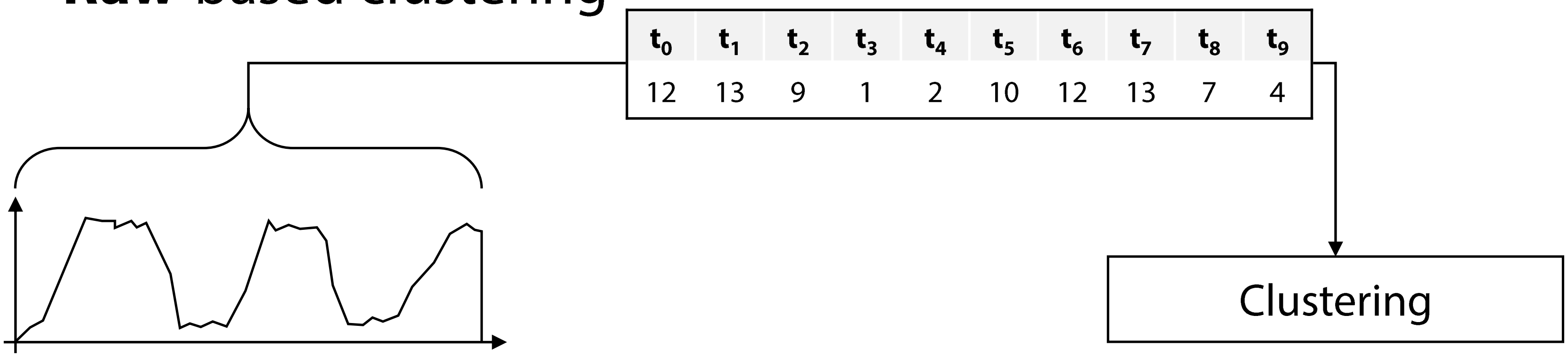
- **Feature**-based clustering



Clustering

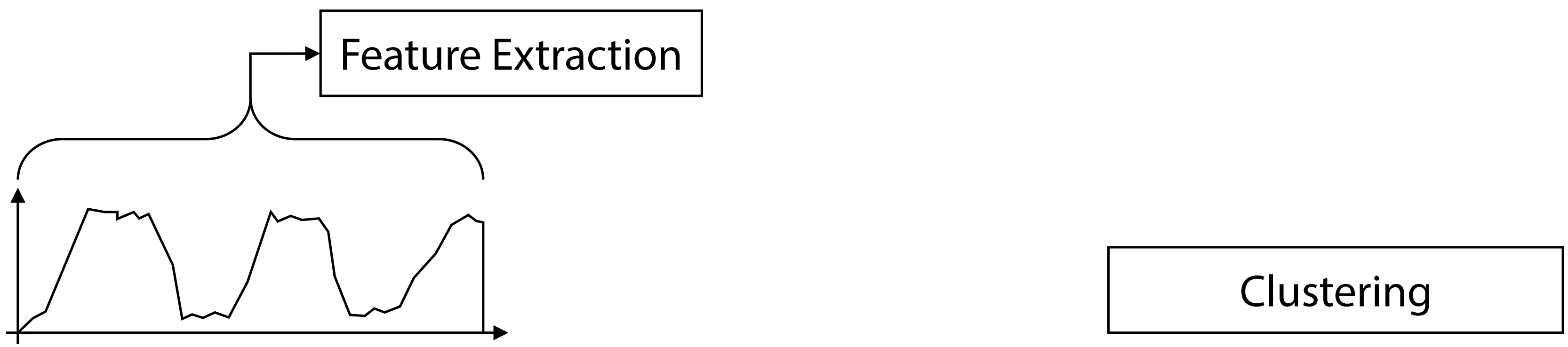
Clustering

- **Raw**-based clustering



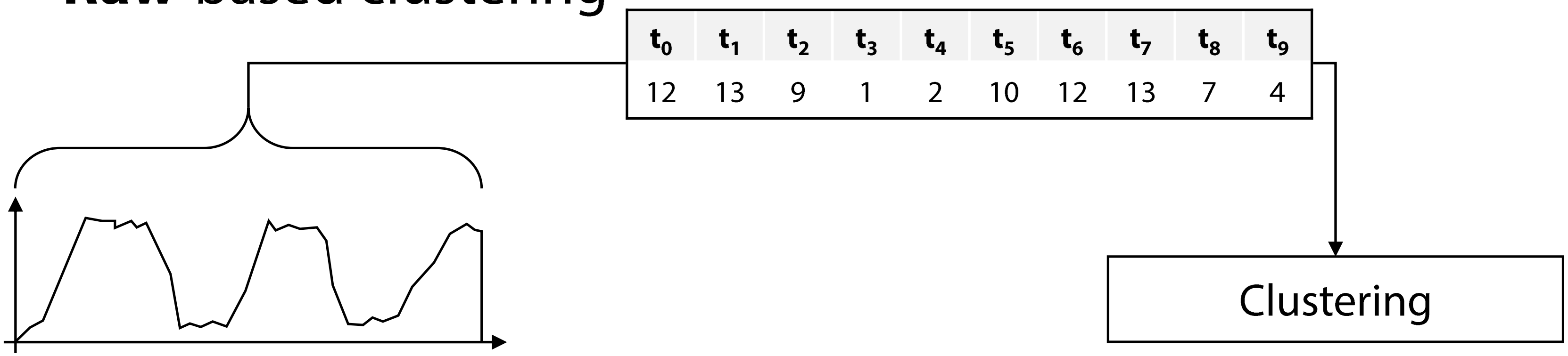
t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9
12	13	9	1	2	10	12	13	7	4

- **Feature**-based clustering

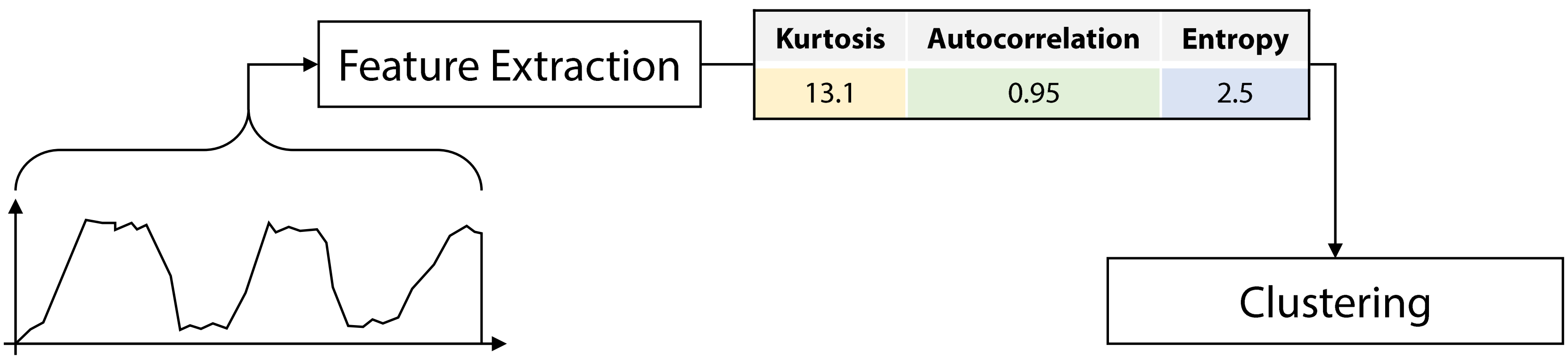


Clustering

- **Raw**-based clustering

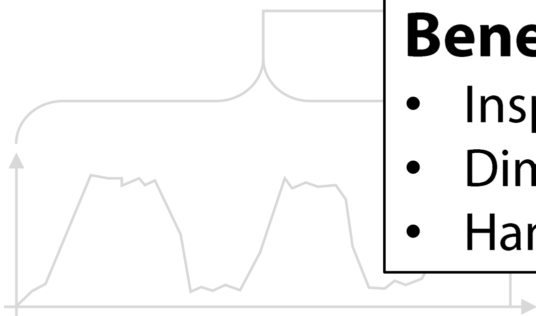


- **Feature**-based clustering



Clustering

- **Raw**-based clustering

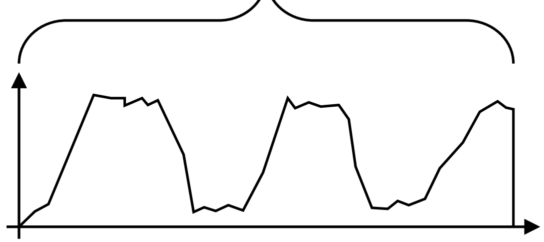


Clustering

Benefits:

- Inspect time series properties
- Dimensionality reduction
- Handle unequal time series lengths

- **Feature**-based clustering



Feature Extraction

Kurtosis	Autocorrelation	Entropy
13.1	0.95	2.5

Clustering

Clustering Methods

- **Method** = triplet of (model, features, variant)

Clustering Methods

- **Method** = triplet of (model, features, variant)

Model	Features	Variant
…	…	…
k-means	raw	no post-processing
BIRCH	feature set A	clip [0, 1]
Agglomerative	feature set B	drop correlated
…	…	…

Clustering Methods

- **Method** = triplet of (model, features, variant)

Model	Features	Variant
…	…	…
k-means	raw	no post-processing
BIRCH	feature set A	clip [0, 1]
Agglomerative	feature set B	drop correlated
…	…	…

- Which one has the best clustering **quality**?

Clustering Methods

- **Method** = triplet of (model, features, variant)

Model	Features	Variant
…	…	…
k-means	raw	no post-processing
BIRCH	feature set A	clip [0, 1]
Agglomerative	feature set B	drop correlated
…	…	…

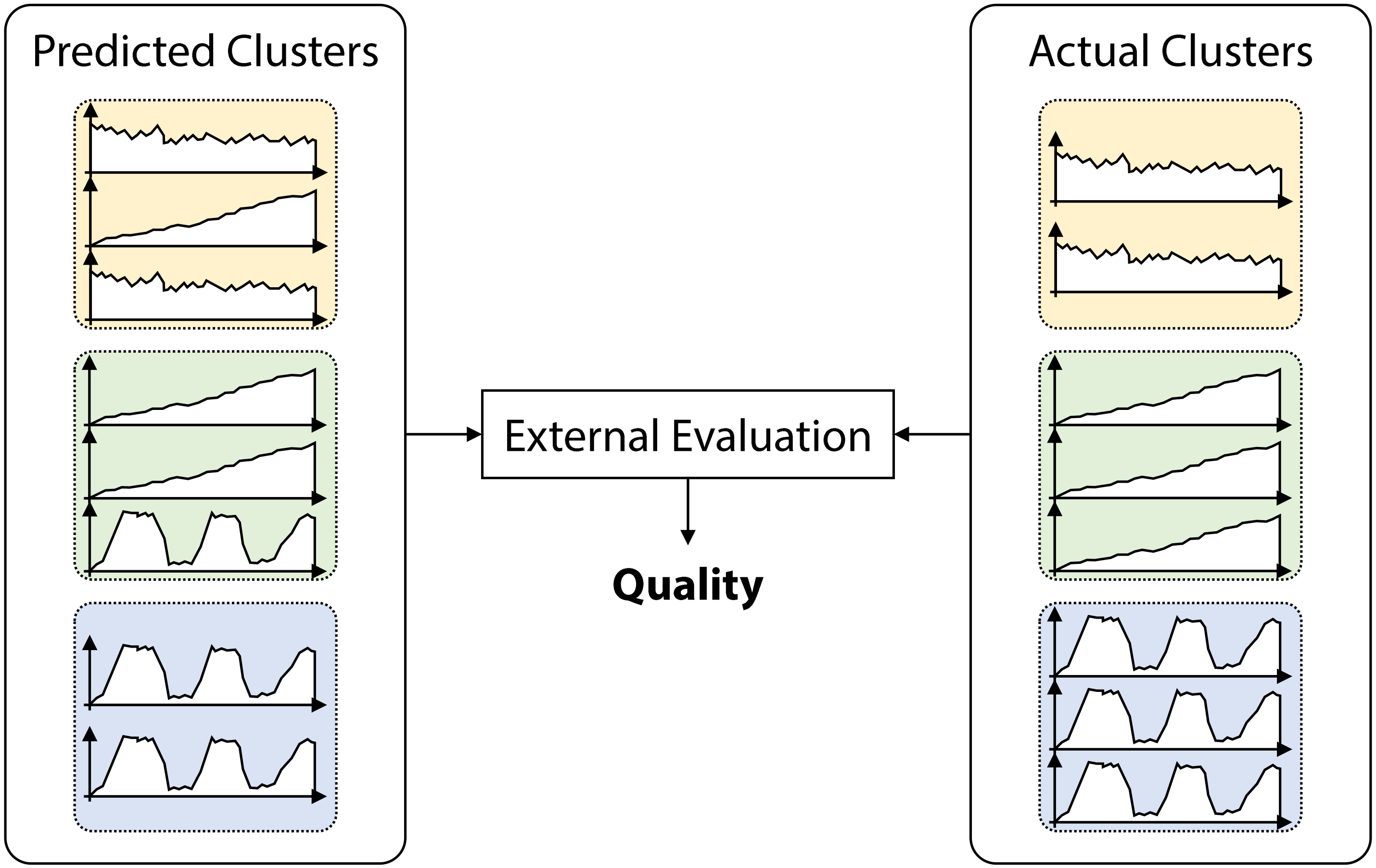
- Which one has the best clustering **quality**?
- What are the run-time **costs**?

Assessing Quality

- Any **external evaluation metric**
- Requirement: **Labeled data**

Assessing Quality

- Any **external evaluation metric**
- Requirement: **Labeled data**



Assessing Costs

- Idea: Use **run-time complexities**

Assessing Costs

- Idea: Use **run-time complexities**
- Problem: **Identical estimates**

```python
def func1(n=1000):
    x = []
    for i in range(n):
        x.append(i)
    return x
```

$O(n)$

```python
from numba import jit

@jit
def func2(n=1000):
    x = []
    for i in range(n):
        x.append(i)
    return x
```

$O(n)$

```python
def func3(n=1000):
    return [i for i in range(n)]
```

$O(n)$

```python
import numpy as np

def func4(n=1000):
    return np.arange(n).tolist()
```

$O(n)$

Assessing Costs

- Idea: Use **run-time complexities**
- Problem: **Identical estimates** may yield **different run times**

```python
def func1(n=1000):
    x = []
    for i in range(n):
        x.append(i)
    return x
```

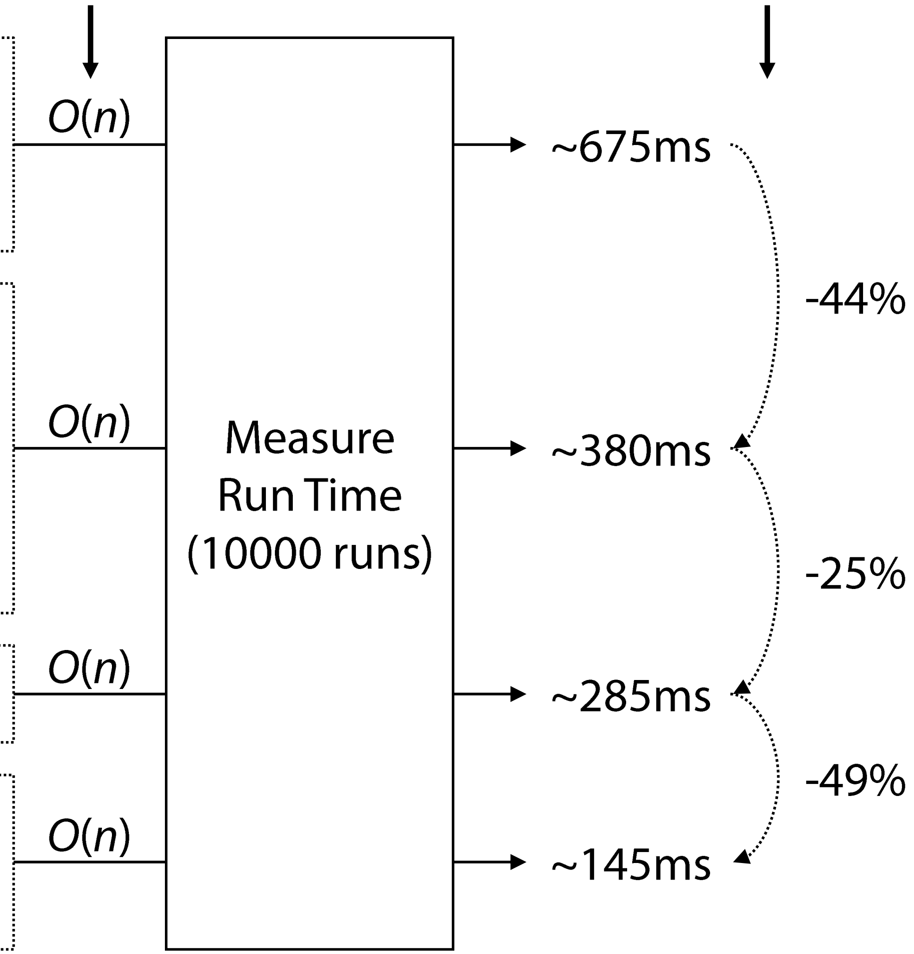
```python
from numba import jit

@jit
def func2(n=1000):
    x = []
    for i in range(n):
        x.append(i)
    return x
```

```python
def func3(n=1000):
    return [i for i in range(n)]
```

```python
import numpy as np

def func4(n=1000):
    return np.arange(n).tolist()
```

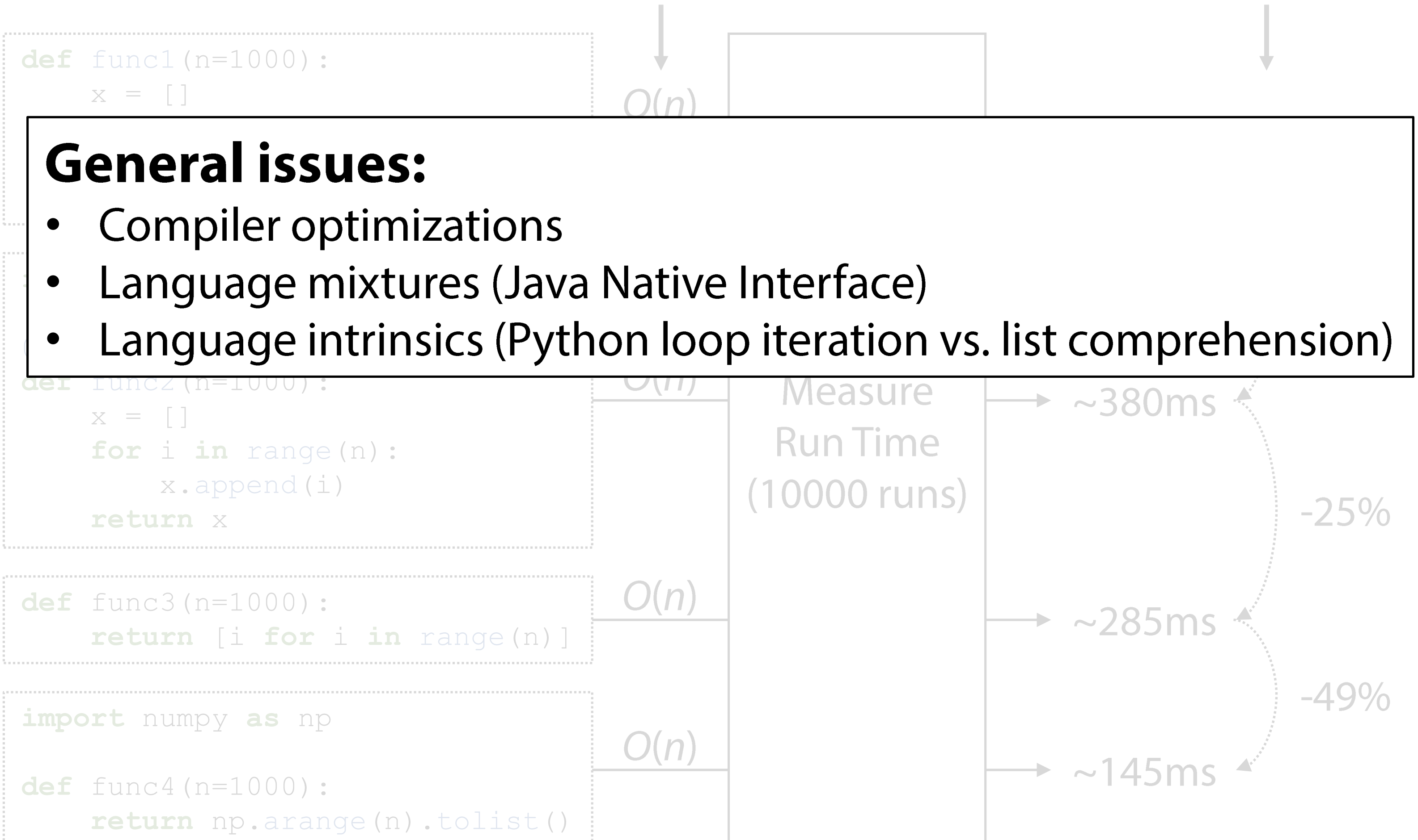


Assessing Costs

- Idea: Use **run-time complexities**
- Problem: **Identical estimates** may yield **different run times**

General issues:

- Compiler optimizations
- Language mixtures (Java Native Interface)
- Language intrinsics (Python loop iteration vs. list comprehension)



Assessing Costs

Measure **actual run time** r on a concrete machine

Assessing Costs

Measure **actual run time** r on a concrete machine

- Given: Set of n time series of length t, sets of p features

$$r_{\text{Method}} \quad =$$

Assessing Costs

Measure **actual run time** r on a concrete machine

- Given: Set of n time series of length t, sets of p features

$$r_{\text{Method}} = \boxed{r_{\text{Model}}}$$

Measure Model Fitting: n, p

Assessing Costs

Measure **actual run time** r on a concrete machine

- Given: Set of n time series of length t, sets of p features

$$r_{\text{Method}} = \boxed{r_{\text{Model}}} + \boxed{r_{\text{Features}}}$$

Measure Model Fitting: n, p

Measure Feature Calculation: n, t

Assessing Costs

Measure **actual run time** r on a concrete machine

- Given: Set of n time series of length t, sets of p features

$$r_{\text{Method}} = r_{\text{Model}} + r_{\text{Features}} + r_{\text{Variant}}$$

Measure Model Fitting: n, p

Measure Feature Calculation: n, t

Measure Variant Calculation: n, p

Assessing Costs

Measure **actual run time** r on a concrete machine

- Given: Set of n time series of length t, sets of p features

$$r_{\text{Method}} = \boxed{r_{\text{Model}}} + \boxed{r_{\text{Features}}} + \boxed{r_{\text{Variant}}}$$

Measure Model Fitting: n, p	Measure Feature Calculation: n, t	Measure Variant Calculation: n, p

- Robust: Measure multiple times → set of measurements R

Assessing Costs

Measure **actual run time** r on a concrete machine

- Given: Set of n time series of length t, sets of p features

$$r_{\text{Method}} = \boxed{r_{\text{Model}}} + \boxed{r_{\text{Features}}} + \boxed{r_{\text{Variant}}}$$

Measure Model Fitting: n, p

Measure Feature Calculation: n, t

Measure Variant Calculation: n, p

- Robust: Measure multiple times → set of measurements R

$$r = \frac{1}{|Q|} \sum_{r' \in Q} r' \qquad Q = \{r' \in R \mid r' \geq q_l(R) \land r' \leq q_u(R)\}$$

Assessing Costs

Measure **actual run time** r on a concrete machine

- Given: Set of n time series of length t, sets of p features

$$r_{\text{Method}} = r_{\text{Model}} + r_{\text{Features}} + r_{\text{Variant}}$$

Measure Model Fitting: n, p

Measure Feature Calculation: n, t

Measure Variant Calculation: n, p

- Robust: Measure multiple times → set of measurements R

$$r = \frac{1}{|Q|} \sum_{r' \in Q} r' \qquad Q = \{r' \in R \mid r' \geq q_l(R) \land r' \leq q_u(R)\}$$

lower quantile

upper quantile

Time Series Characteristics (TSC)

Group	Subgroup	#Features
Distributional	Dispersion	3
	Dispersion (blockwise)	10
	Duplicates	5
	Distribution	16
Temporal	Dispersion	2
	Dispersion (blockwise)	10
	Similarity	17
	Frequency	17
	Linearity	44
Complexity	Entropy	13
	Complexity (miscellaneous)	5
	Flatness	15
	Peaks	8
Statistical Tests	-	2

Time Series Characteristics (TSC)

Group	Subgroup	#Features
Distributional	Dispersion	3
	Dispersion (blockwise)	10
	Duplicates	5
	Distribution	16
Temporal	Dispersion	2
	Dispersion (blockwise)	
	Similarity	
	Frequency	
	Linearity	44
Complexity	Entropy	13
	Complexity (miscellaneous)	5
	Flatness	15
	Peaks	8
Statistical Tests	-	2

4 main groups
13 subgroups

Time Series Characteristics (TSC)

Group	Subgroup	#Features
Distributional	Dispersion	3
	Dispersion (blockwise)	10
	Duplicates	5
	Distribution	16
	Dispersion	2
		10
		17
		17
	Linearity	44
Complexity	Entropy	13
	Complexity (miscellaneous)	5
	Flatness	15
	Peaks	8
Statistical Tests	-	2

43 characteristics
167 features with parameterization

Evaluation

- Data: **UCR** time series classification archive
  - 128 datasets
  - Various domains (synthetic, sensors, motion, image, ECG, etc.)

Evaluation

- Data: **UCR** time series classification archive
  - 128 datasets
  - Various domains (synthetic, sensors, motion, image, ECG, etc.)
- External evaluation metric: **ARI** (adjusted Rand index)

Evaluation

- Data: **UCR** time series classification archive
  - 128 datasets
  - Various domains (synthetic, sensors, motion, image, ECG, etc.)
- External evaluation metric: **ARI** (adjusted Rand index)
- Run-time measurement: **30** runs, quantile range [**0.1**, **0.9**]

Evaluation: Methods

- **Models:**

Evaluation: Methods

- **Models:**
  - k-means
  - BIRCH
  - Agglomerative clustering (Ward's linkage + Euclidean distance)
  - Agglomerative clustering (weighted average linkage + Euclidean dist.)
  - Agglomerative clustering (weighted average linkage + cosine distance)

Evaluation: Methods

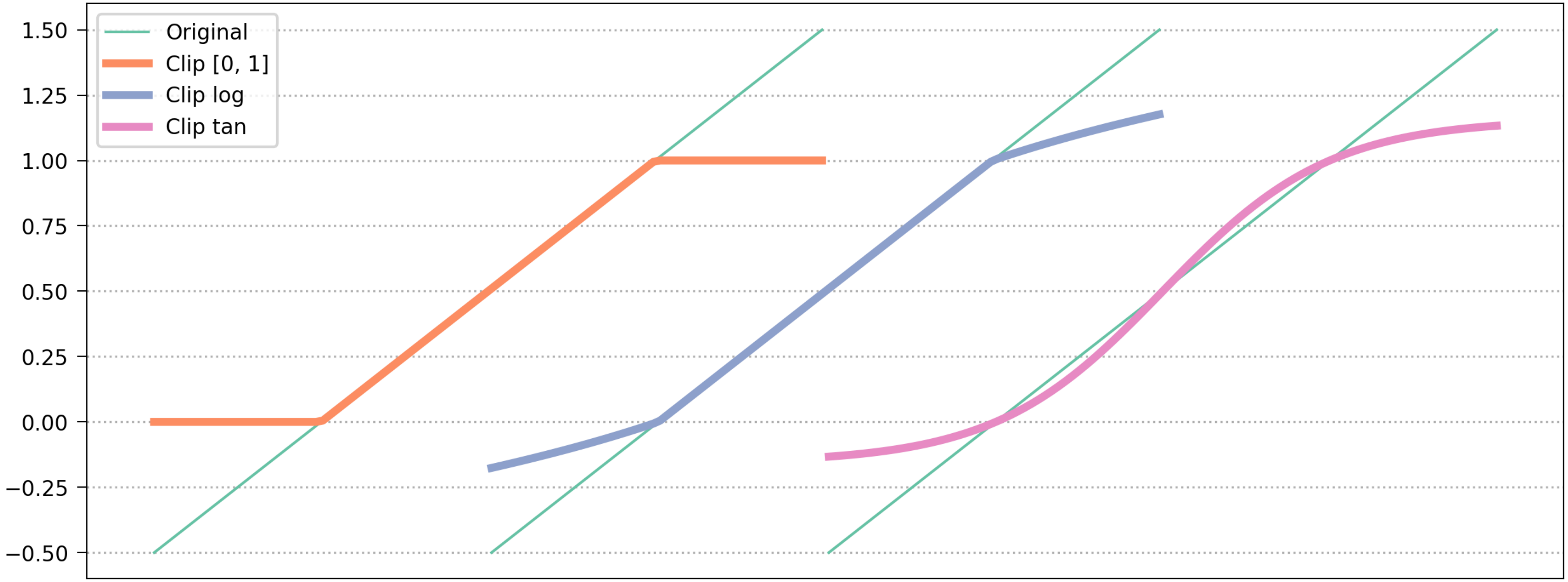
- **Models:**
  - k-means
  - BIRCH
  - Agglomerative clustering (Ward's linkage + Euclidean distance)
  - Agglomerative clustering (weighted average linkage + Euclidean dist.)
  - Agglomerative clustering (weighted average linkage + cosine distance)
- **Features:**
  - 4 main groups + 13 subgroups + all TSC
  - Raw time series data

Evaluation: Methods

- **Models:**
  - k-means
  - BIRCH
  - Agglomerative clustering (Ward's linkage + Euclidean distance)
  - Agglomerative clustering (weighted average linkage + Euclidean dist.)
  - Agglomerative clustering (weighted average linkage + cosine distance)
- **Features:**
  - 4 main groups + 13 subgroups + all TSC
  - Raw time series data
- **Variants** (for features):
  - Dropping correlated features
  - Clipping to [0, 1] + logarithm-based clipping + tangent-based clipping
  - All combinations of dropping + clipping variants
  - No post-processing

Evaluation: Methods

- **Models:**
  - k-means



  - Clipping to [0, 1] + logarithm-based clipping + tangent-based clipping
  - All combinations of dropping + clipping variants
  - No post-processing

Evaluation: Methods

- **Models:** 5 models
  - k-means
  - BIRCH
  - Agglomerative clustering (Ward's linkage + Euclidean distance)
  - Agglomerative clustering (weighted average linkage + Euclidean dist.)
  - Agglomerative clustering (weighted average linkage + cosine distance)
- **Features:** 18 feature sets + raw
  - 4 main groups + 13 subgroups + all TSC
  - Raw time series data
- **Variants** (for features): 8 variants
  - Dropping correlated features
  - Clipping to [0, 1] + logarithm-based clipping + tangent-based clipping
  - All combinations of dropping + clipping variants
  - No post-processing

Evaluation: Methods

- **Models:**
  - k-means
  - BIRCH
  - Agglomerative clustering (Ward's linkage + Euclidean distance)
  - Agglomerative clustering (weighted average linkage + Euclidean dist.)
  - Agglomerative clustering (weighted average linkage + cosine distance)
- **Features:**
  - 4 main groups + 13 subgroups + all TSC
  - Raw time series data
- **Variants** (for features):
  - Dropping correlated features
  - Clipping to [0, 1] + logarithm-based clipping + tangent-based clipping
  - All combinations of dropp
  - No post-processing

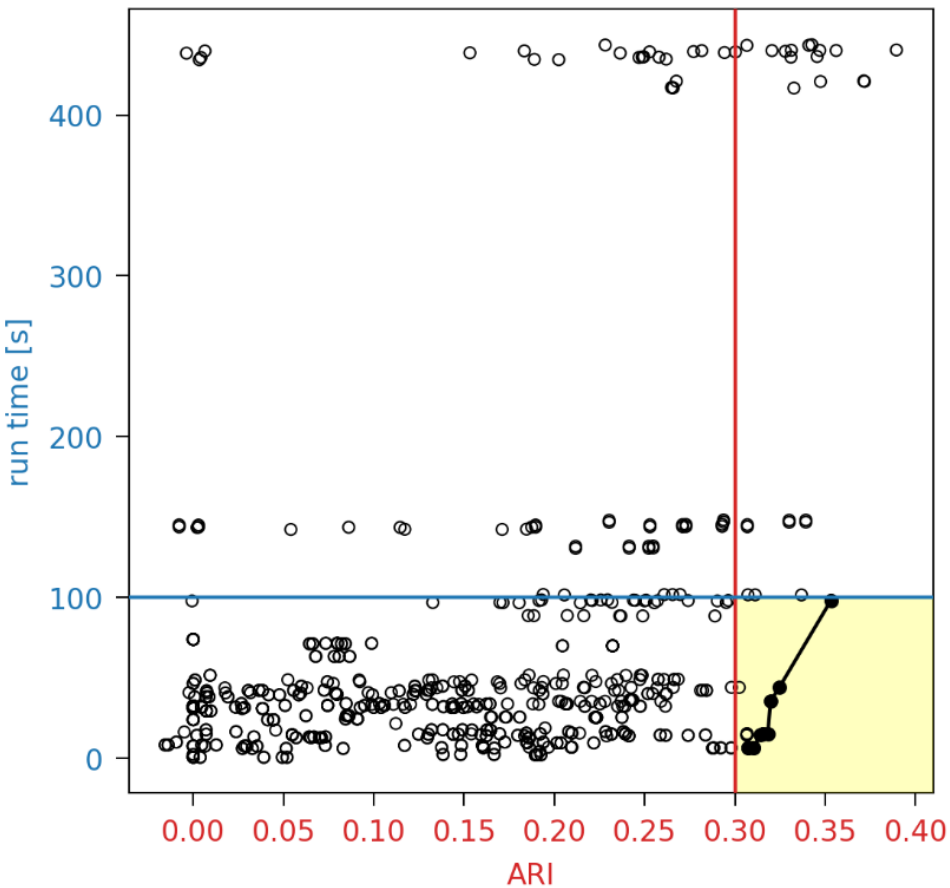
5 models

18 feature sets + raw

8 variants

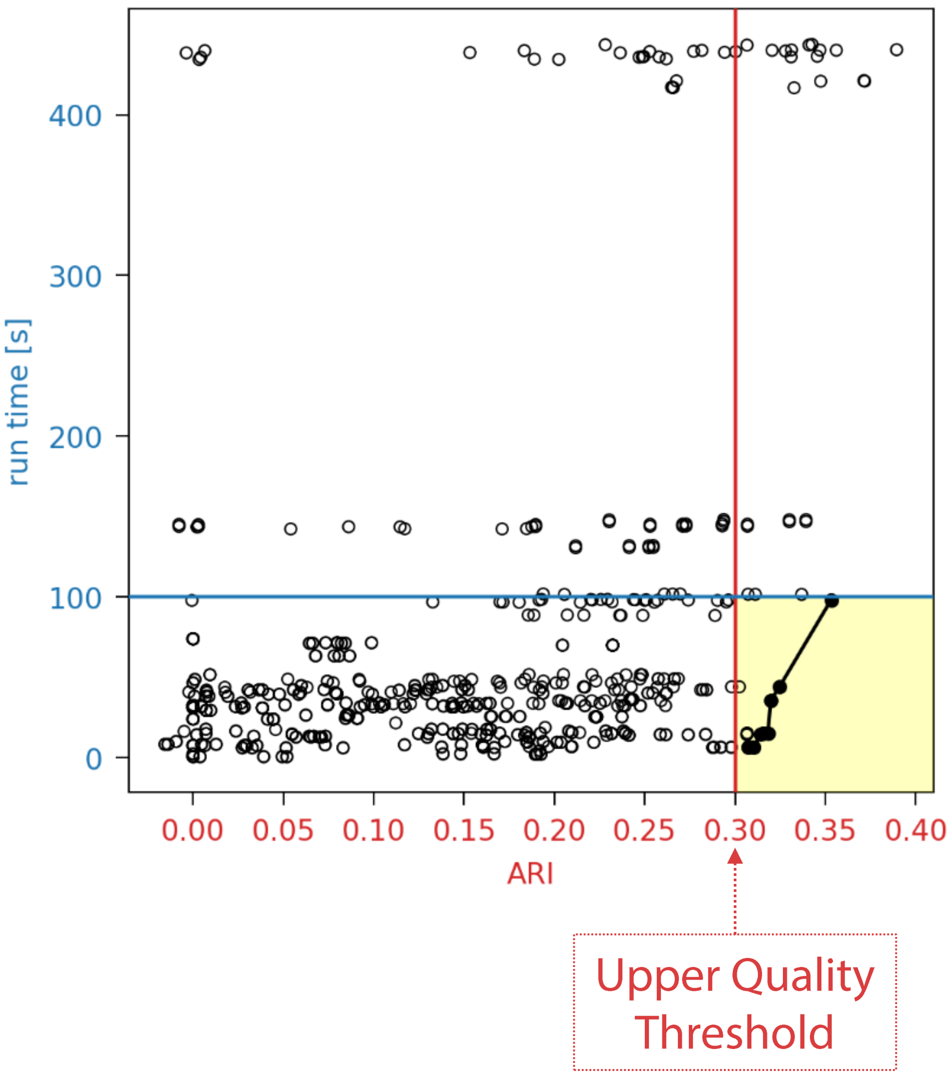
$5 \cdot 18 \cdot 8 + 5 =$ **725 methods**

Results: Quality-Cost-Trade-off



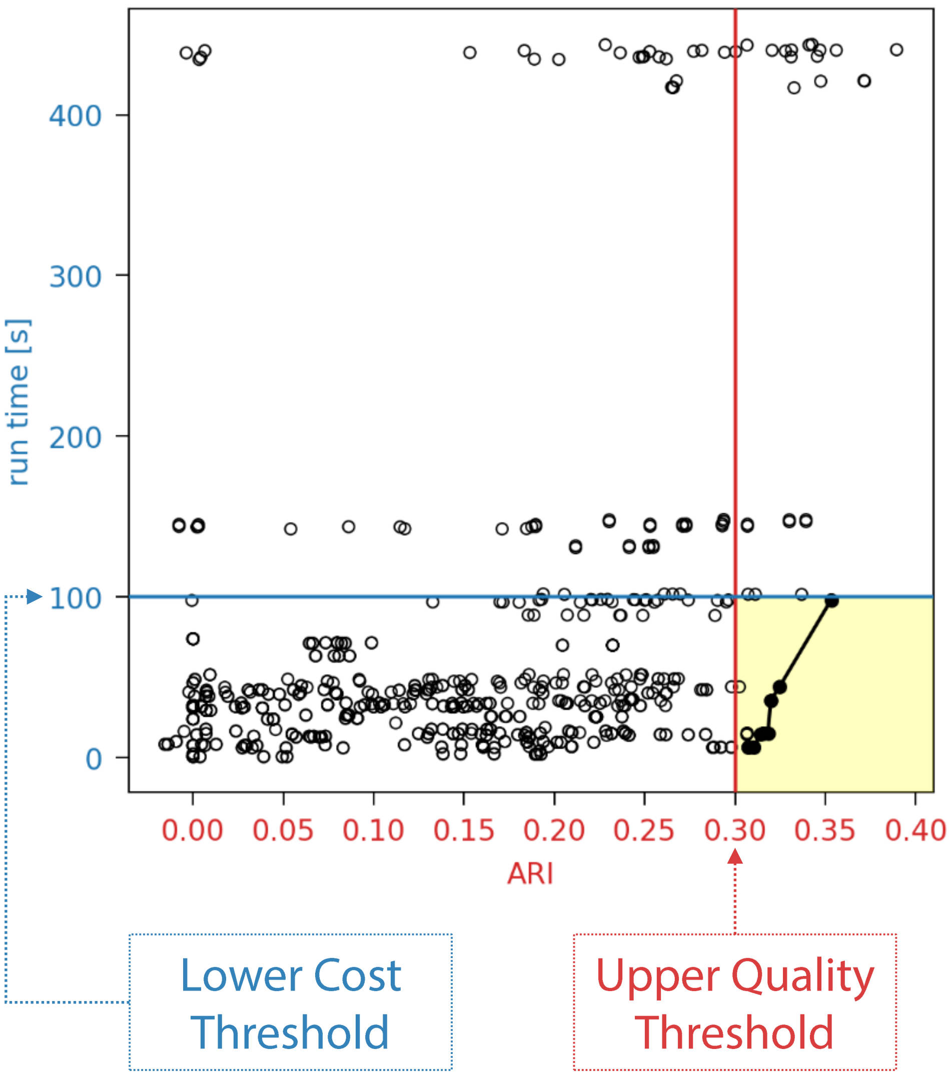
Dataset:	ElectricDevices
Type:	Device
Samples:	16637
Time series length:	96

Results: Quality-Cost-Trade-off



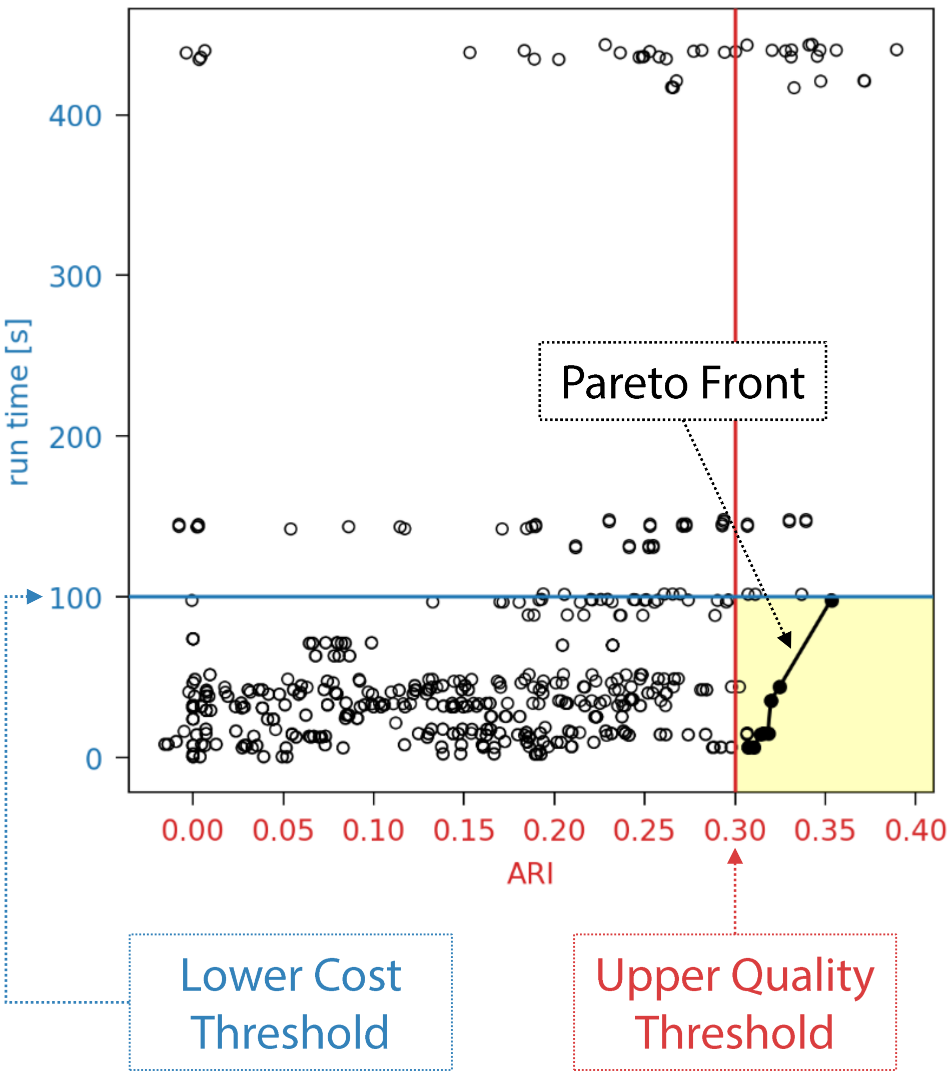
Dataset:	ElectricDevices
Type:	Device
Samples:	16637
Time series length:	96

Results: Quality-Cost-Trade-off



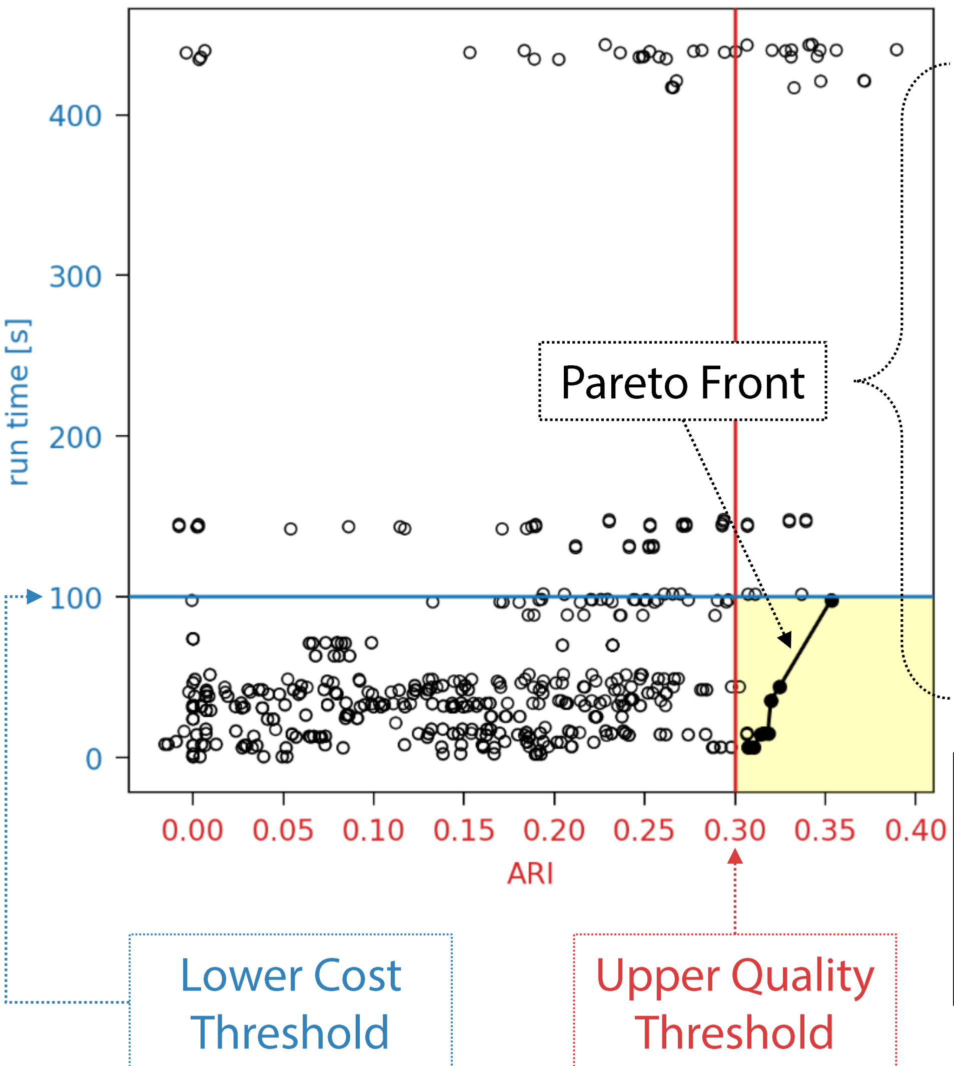
Dataset:	ElectricDevices
Type:	Device
Samples:	16637
Time series length:	96

Results: Quality-Cost-Trade-off



Dataset:	ElectricDevices
Type:	Device
Samples:	16637
Time series length:	96

Results: Quality-Cost-Trade-off

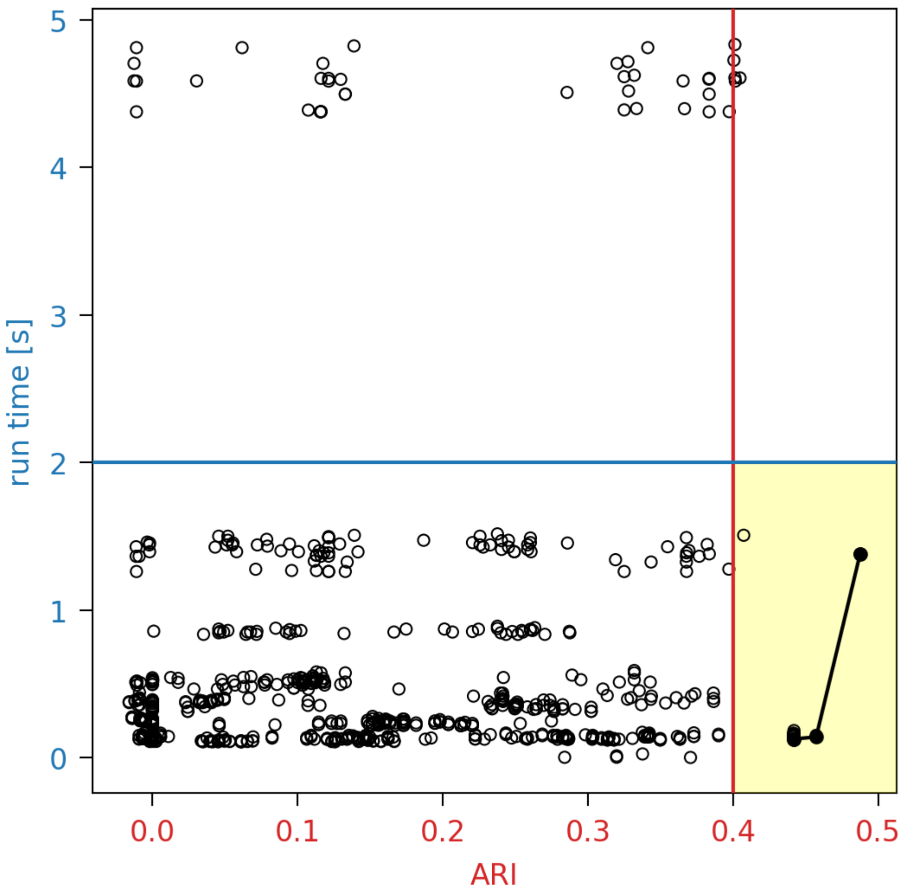


Model	Features	Variant	ARI	Run time [s]
`l`	`complexity`		0.35	98.25
`l`	`c_entropy`	`01_d`	0.32	44.10
`k`	`c_entropy`	`log_d`	0.32	35.51
`l`	`d_dispersion_b`	`log`	0.32	15.30
`l`	`d_dispersion_b`	`01`	0.32	15.27
`l`	`t_dispersion_b`	`01`	0.31	14.49
`k`	`t_dispersion_b`	`tan_d`	0.31	6.55
`k`	`t_dispersion_b`	`tan`	0.31	6.53
`k`	`d_dispersion_b`	`log`	0.31	6.38
`k`	`d_dispersion_b`	`01_d`	0.31	6.38
`k`	`d_dispersion_b`	`01`	0.31	6.36

Dataset: ElectricDevices
Type: Device
Samples: 16637
Time series length: 96

Models: `l` = linkage, `k` = m-means. Features: `group_subgroup` (group abbreviated to first letter), `_b` = blockwise. Variants: empty = no post-processing, `01` = clip [0, 1], `log` = logarithm-based clipping, `tan` = tangent-based clipping, `_d` = variant + dropping correlated features.

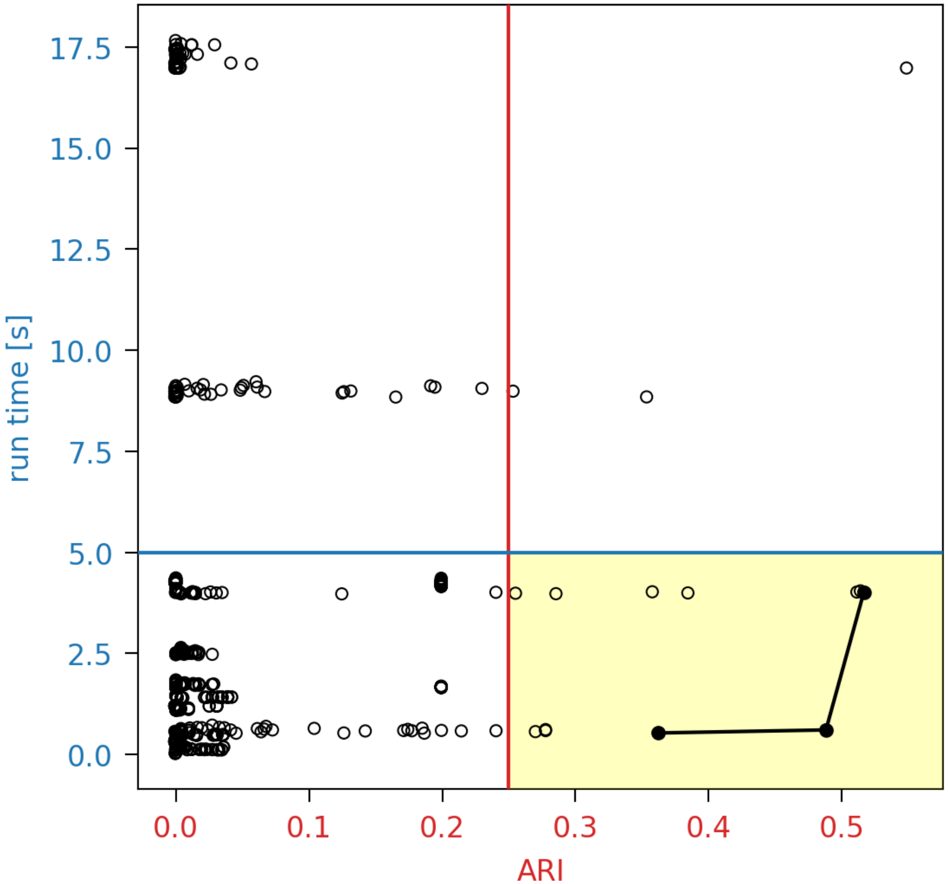
Results: Quality-Cost-Trade-off



Model	Features	Variant	ARI	Run time [s]
k	temporal	d	0.49	1.38
l	t_dispersion_b	log	0.46	0.14
l	c_flatness	01	0.44	0.13
l	c_flatness		0.44	0.13

Dataset:	FaceFour
Type:	Image
Samples:	112
Time series length:	350

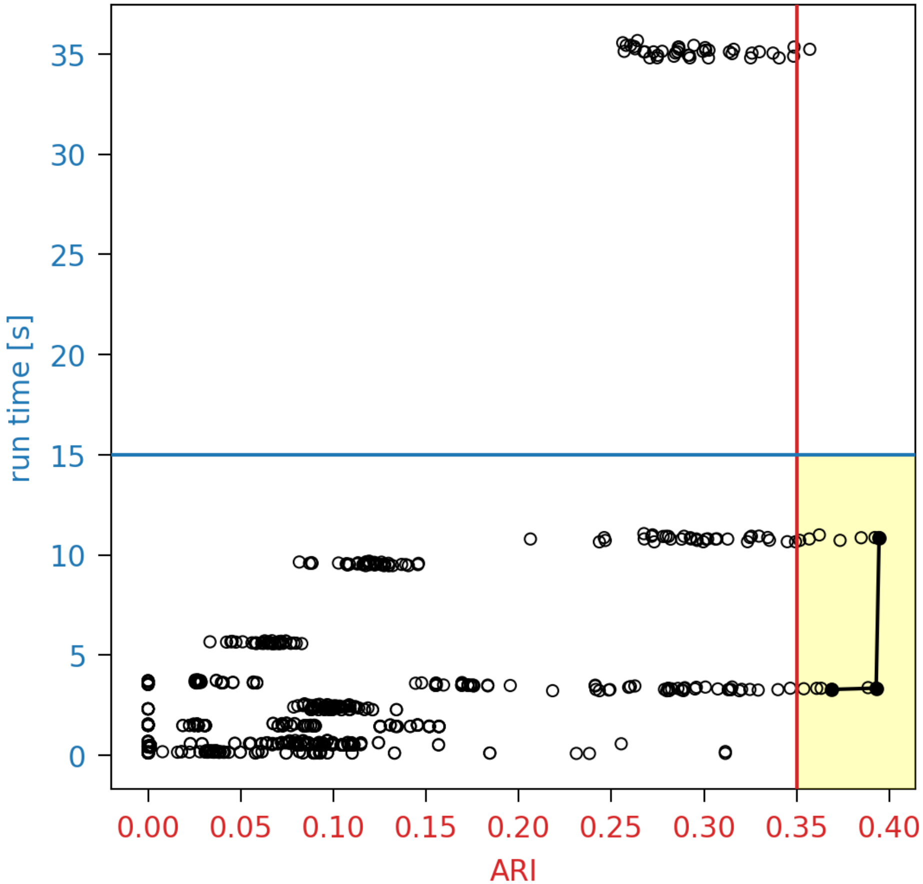
Results: Quality-Cost-Trade-off



Model	Features	Variant	ARI	Run time [s]
k	t_similarity	01_d	0.52	4.02
b	t_linearity		0.49	0.61
lw	t_linearity	01	0.36	0.53

Dataset:	ItalyPowerDemand
Type:	Sensor
Samples:	1096
Time series length:	24

Results: Quality-Cost-Trade-off



Model	Features	Variant	ARI	Run time [s]
b	temporal	01_d	0.39	10.87
b	t_linearity	log	0.39	3.35
b	t_linearity	01	0.37	3.29

Dataset:	FiftyWords
Type:	Image
Samples:	905
Time series length:	270

Conclusion

- Clustering **method selection** based on actual run-time costs:
  - Models
  - Features
  - Variants

Conclusion

- Clustering **method selection** based on actual run-time costs:
  - Models
  - Features
  - Variants
- User can selected method via **quality-cost trade-off**

Conclusion

- Clustering **method selection** based on actual run-time costs:
    - Models
    - Features
    - Variants
- User can selected method via **quality-cost trade-off**
- Future work: Apply to **other areas:**
    - Classification
    - Forecasting
    - …

Selecting Time Series Clustering Methods based on Run-Time Costs

Andreas Schörgenhumer
Paul Grünbacher
Hanspeter Mössenböck

12.11.2020

JOHANNES KEPLER UNIVERSITY LINZ

TSC: Distributional

Subgroup	Characteristic	Description
Dispersion	`kurtosis`	measure of tailedness
	`skewness`	measure of asymmetry
	`shift`	mean minus the median of those values that are smaller than the mean
Dispersion (blockwise)	`lumpiness`	variance of the variances of blocks
	`stability`	variance of the mean of blocks
Duplicates	`normalized_duplicates_max`	number of duplicates that have the maximum value of the data
	`normalized_duplicates_min`	number of duplicates that have the minimum value of the data
	`percentage_of_reoccurring_datapoints`	number of unique duplicates compared to the number of unique values
	`percentage_of_reoccurring_values`	number of duplicates compared to the length of the data
	`percentage_of_unique_values`	number of unique values compared to the length of the data
Distributio n	`quantile`	threshold below which x% of the ordered values of the data are, giving a hint on the distribution
	`ratio_beyond_r_sigma`	ratio of values that are more than a factor $r \cdot \sigma$ away from the mean
	`ratio_large_standard_deviation`	ratio between the standard deviation and the (max – min) range of the data (based on the "range rule of thumb")

TSC: Temporal

Subgroup	Characteristic	Description
Dispersion	`mean_abs_change`	average absolute difference of two consecutive values
	`mean_second_derivative_central`	measure of the rate of the rate of change
Dispersion (blockwise)	`level_shift`	maximum difference in mean between consecutive blocks
	`variance_change`	maximum difference in variance between consecutive blocks
Similarity	`hurst`	measure of long-term memory of a time series, related to auto-correlation
	`autocorrelation`	correlation of a signal with a lagged version of itself
Frequency	`periodicity`	power (intensity) of specified frequencies in the signal (based on the periodogram)
	`agg_periodogram`	results of user-defined aggregation functions (e.g., fivenum) calculated on the periodogram
Linearity	`linear_trend_slope`	measure of linearity: slope
	`linear_trend_rvalue2`	measure of linearity: r^2 (coefficient of determination)
	`agg_linear_trend_slope`	variance-aggregated slopes of blocks
	`agg_linear_trend_rvalue2`	mean-aggregated r^2 of blocks
	`c3`	measure of non-linearity (originally from the physics domain)
	`time_reversal_asymmetry_statistic`	asymmetry of the time series if reversed, which can be a measure of non-linearity

TSC: Complexity

Subgroup	Characteristic	Description
Entropy	`binned_entropy`	fast entropy estimation based on equidistant bins
	`kullback_leibler_score` (KL score)	maximum difference of KL divergences between consecutive blocks, where the KL divergence is a measure of how two probability distributions differ
	`index_of_kullback_leibler_score`	relative location where the maximum KL score was found
Complexity (misc.)	`cid_ce`	measure of complexity invariance
	`permutation_analysis`	measure of complexity through permutation
	`swinging_door_compression_rate`	compression ratio of the signal under a given error tolerance ϵ
Flatness	`normalized_crossing_points`	number of times a time series crosses the mean line (based on fickleness)
	`normalized_above_mean`	number of values that are higher than the mean
	`normalized_below_mean`	number of values that are lower than the mean
	`normalized_longest_strike_above_mean`	relative length of the longest series of consecutive values above the mean
	`normalized_longest_strike_below_mean`	relative length of the longest series of consecutive values below the mean
	`flat_spots`	maximum run-length of values when divided into quantile-based bins
Peaks	`normalized_number_peaks`	number of peaks, where a peak of support n is defined as a value which is bigger than its n left and n right neighbors
	`step_changes`	number of times the time series significantly shifts its value range

TSC: Statistical Tests

Subgroup	Characteristic	Description
-	`adf`	augmented Dickey-Fuller (ADF) test for unit root presence
	`kpss`	Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test for stationarity