Table of Contents

1 Machine learning in Time Series Analysis

1.1 The tasks of ML in TS

1.1.1 Introduction in ML for TS

In the terms of Machine learning the Time Series Analysis can be divided into:

- supervised tasks and methods (Point Estimates, Probabilistic Distribution Estimation).
- unsupervised tasks and methods.
- other (such as semi- or self- supervised and so-on).

The supervised task assume that we have time series where data are can be called labeled. The typical tasks of the supervised models are:

- regression
  - forecast;
  - parameters estimation;
  - supervised data filtration;
  - supervised feature extraction and selection;
- classification (of some series patterns, or series behavior its self).
  - pattern classification;
  - supervised anomaly of other events classification;
  - full time series classification.

> *Note*
>
> The parameters estimation task can be considered as supervised if its routine is based on the labeled dataset or *a priory* known parametric model. In the other case (for non-parametric model) parameters estimation task can be considered as unsupervised task.

The unsupervised time series processing task are searching and uncovering undetected patterns in a dataset with no pre-existing labels as, for example, cluster analysis or data-decomposition and data compression algorithms.

The the most popular tasks in the unsupervised time series learning are:

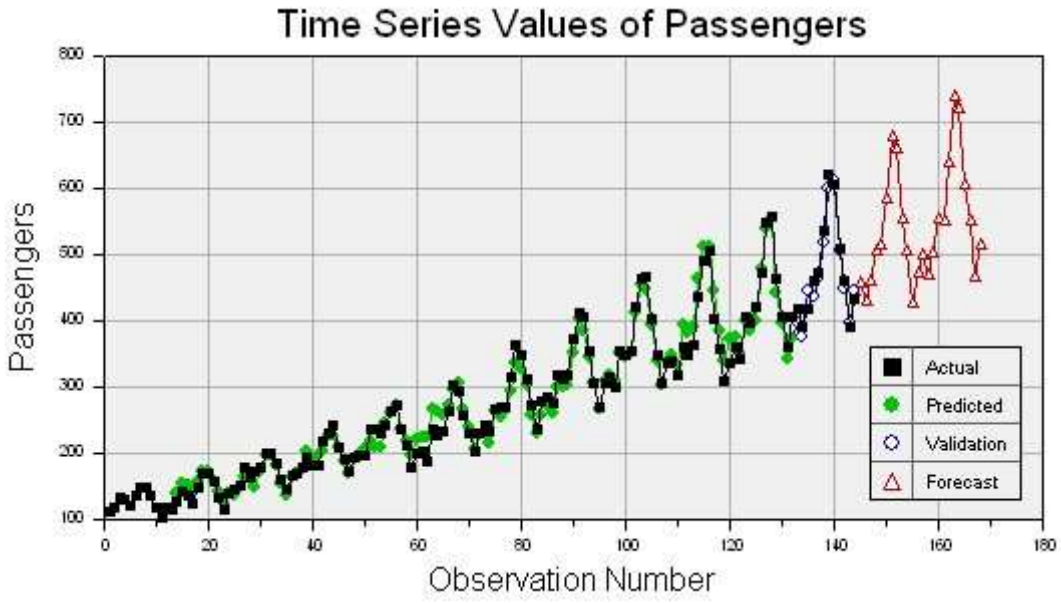
- **anomaly detection** (such as outliers, skipped data, abnormal patterns or any other).
- **feature extraction** or **unsupervised time series decomposition** (trend-seasonality decomposition, PCA, other component decomposition);
- **time series denoisng** (in particular in the case of white Gaussian noises);
- **data clustering**, patterns searching, data segmentation and other.

Pleas note

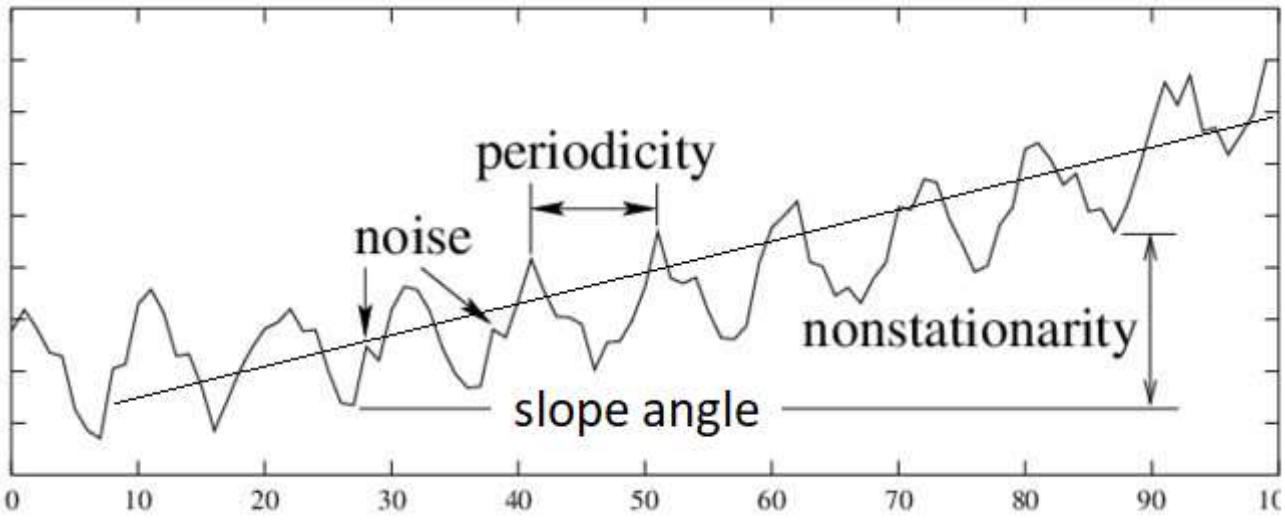
- In some cases it is necessary to eliminate some part of the series that are considered as noise or inferences - this task can be considered as **supervised denoisng or supervised time series filtration**.
- In the case of the series decomposition is performed based on the some *a priory* known model it should be related to supervised learning tasks (in particular supervised denoisng is the example of such decomposition).
- The anomaly detection can be also performed as supervised task if data will be labeled as normal or anomaly or as semi-supervised task if only normal pattern will be labeled.

1.1.2 Examples of ML in TS

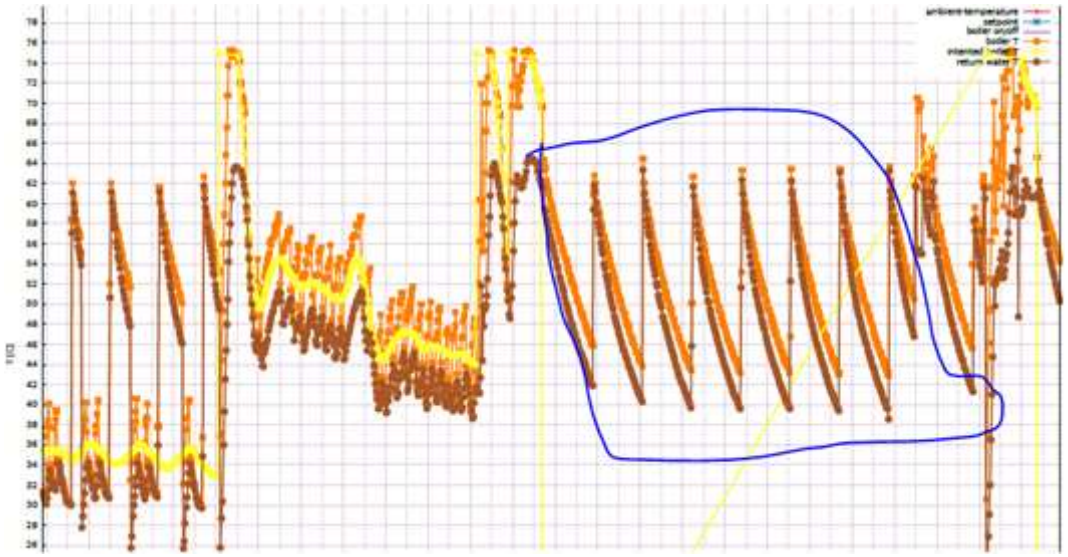
Time series forecast (*supervised learning*) example



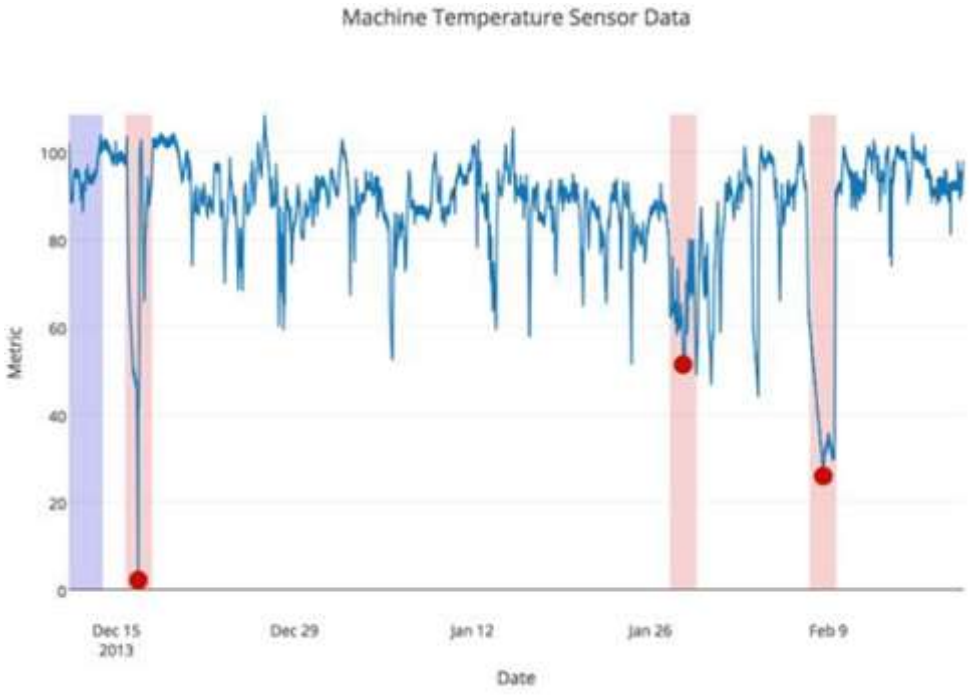
Series parameters estimation example (supervised learning) for parametric *a priory* known model or if estimation performed based on the labeled data-set (in data-driven manner).



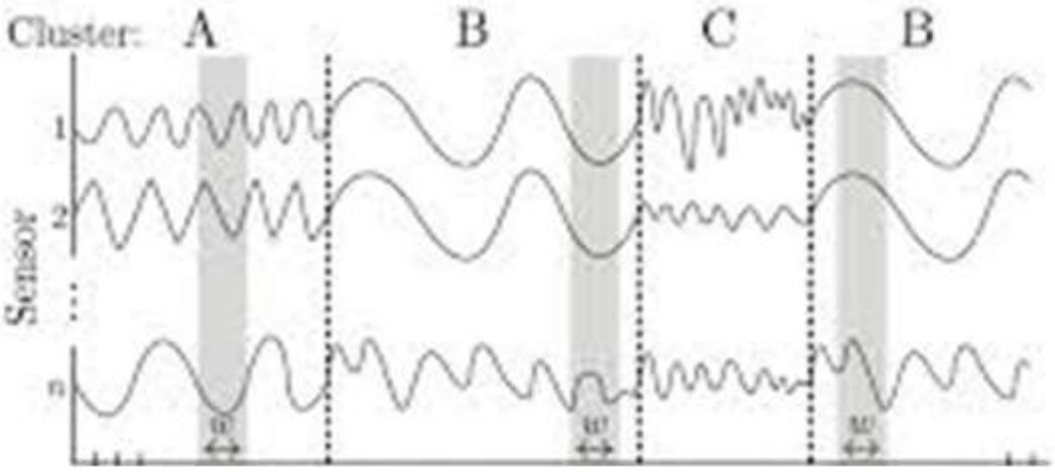
Data or data-pattern (segment of data) classification (*supervised learning*).



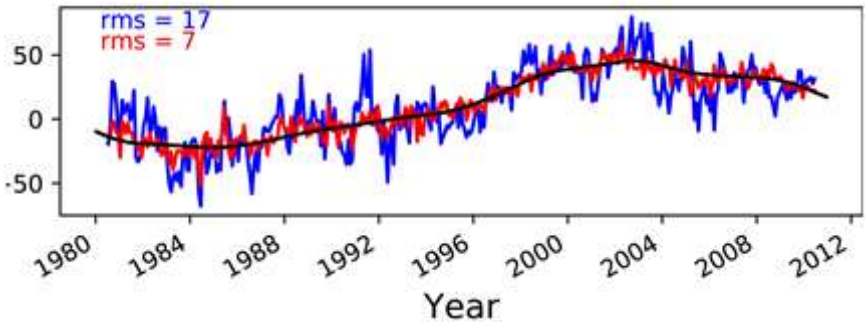
Anomaly detection (outliers, skipped data, abnormal patterns or abnormal behavior, in *general unsupervised learning*).



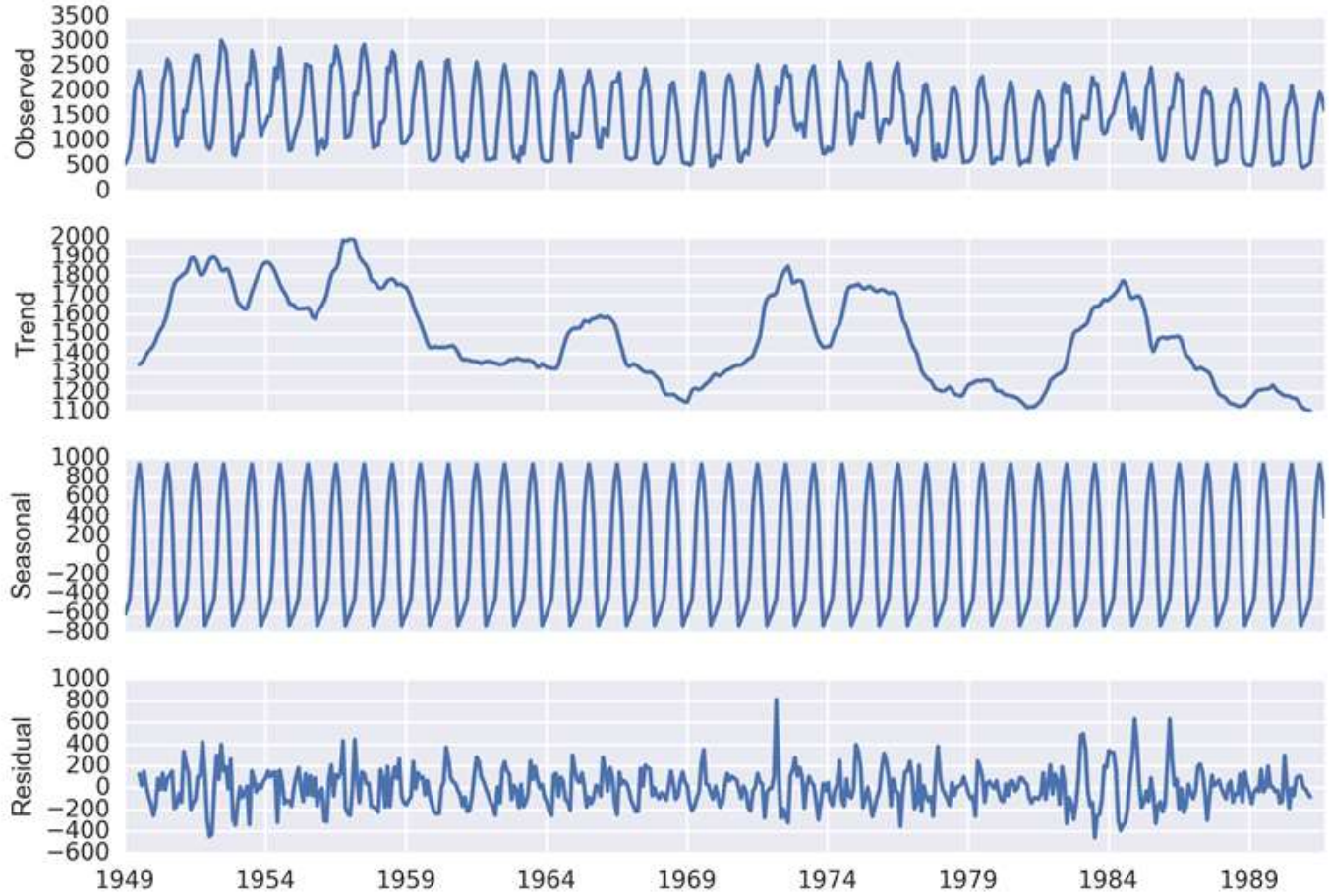
Time series clustering, segmentation, pattern searching (*unsupervised learning*).



Time series denoising from white gaussian noises (*unsupervised learning case*).



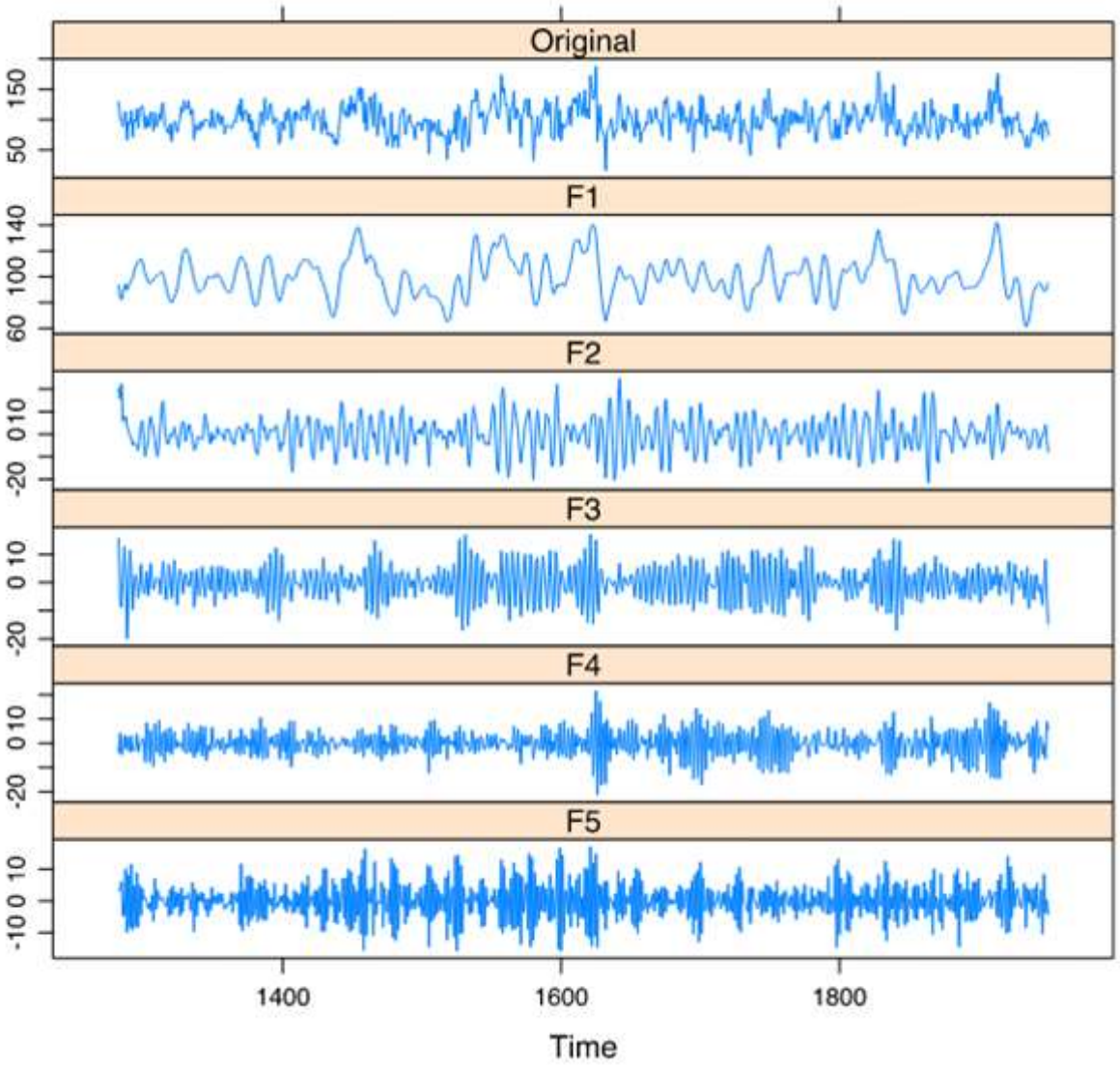
Time series decomposition on obvious trend, seasonal part and residual part (*in general unsupervised learning*).



Time series decomposition without any obvious time series patterns (*in genenral unsupervised learning*),

- If we will rest only slow-changeable components (or for instance, components with the highest variance), than we can resampling data (down sampling) in this case it will be **data compression** task (*unsupervised learning*),
- if we will know by some model or other source of information which components are valuable we can reconstruct series without inferences

and noises (make **time series filtration**) - in this case it will be *supervised learning*.



1.2 Time Series Clustering

1.2.1 Tasks of clustering

Large amount of data stored in serialized databases make the process of its parsing and screening useful information (visualization) i.e. its clustering

actual task.

For instance,

- **Time series database (TSDB),**
- **sensors measurements results,**
- **speech or music pattern searching.**

The process of separating groups according to similarities of data is called clustering.

The goal of clustering is to identify structure in an unlabeled dataset by objectively organizing data into homogeneous groups where the within-group-object similarity is minimized and the between-group-object dissimilarity is maximized.

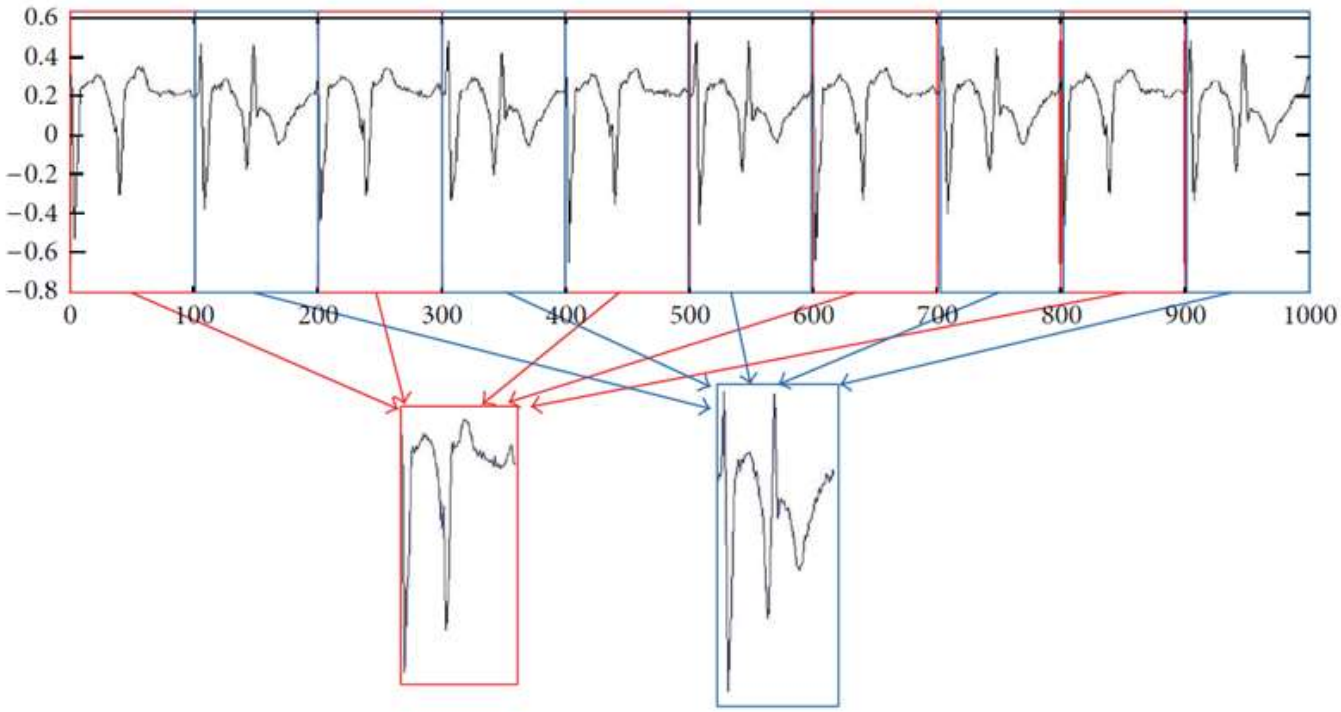
The clustering (or partitioning the dataset of series on a several clusters) consists in the following.

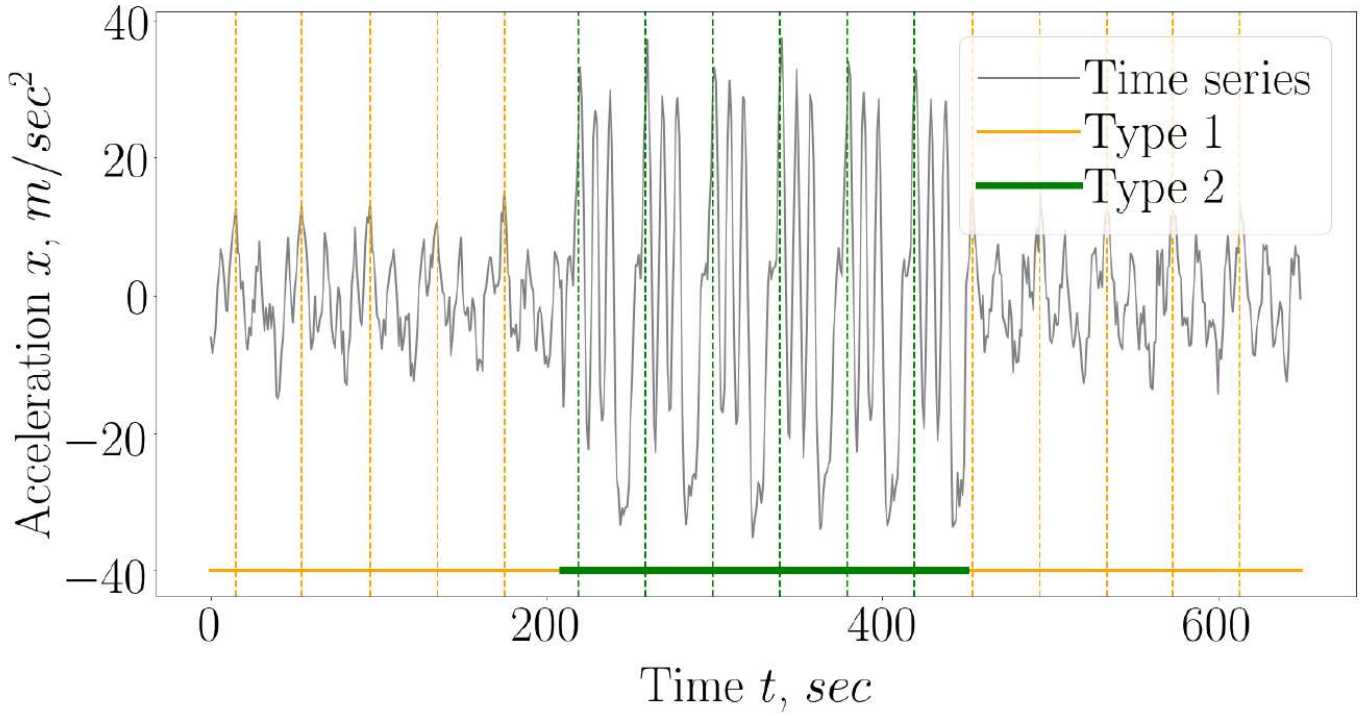
- Given a set of n samples of unlabeled data,
- a clustering method constructs k partitions of the data,

where each partition illustrates a cluster containing at least one object (if $k < n$).

- try to search the centroid of the cluster.
- The partition can be crisp if each object belongs to exactly one cluster, or fuzzy if one object is allowed to be in more than one cluster to a different degree.

Example of time series segments clustering





It can be distinguished a several approaches for clustering.

1. **Raw series clustering.**
2. **Features of the series clustering** (feature-based clustering).
3. **Model of the series clustering** (model-based clustering).

Feature-based clustering Feature extraction here mean that we extract such series segments metrics and features as:

Point-value features

- mean value,
- std values,
- area under curve,
- Autocorrelation or Partial AR lags values,

- main frequencies (in spectrum domain, or pseudo-spectrum domain, obtained by some time to frequency transformation like wavelet or cosine transforms),

Vector-value features

- segments of the spectrum
- features obtained after PCA or other dimension reduction method, like Perceptually Important Points (PIP) or Piecewise Linear Approximation (PLA).
- other non-linear features, for instance, feature vector can be obtained after autoencoder learning or other unsupervised and semi-supervised techniques.
- series coding)like bag of symbols, SFA symbols, for instance).
- feature combinations (bag of features approach).

Model-based clustering

Actually it is similar to feature extraction, but here we need to choose some prelimeniar parameters.

We can estimate first the parameters of the model for one segment on try to analyze other segments or we can first estimate the model for the full

dataset and then analyze each part we divide the data.

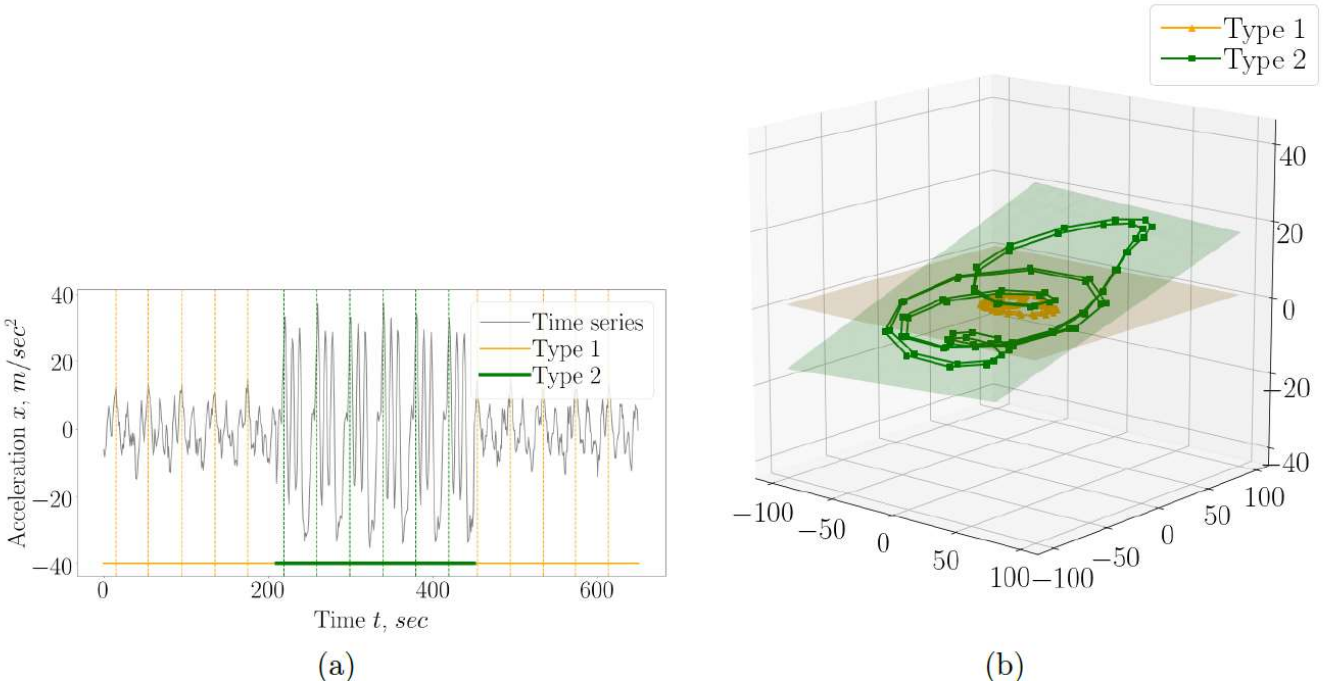
Modeling here mean that we extract such series segments parameters as:

- AR, MA (ARMA) model parameters,
- frequencies filtering, adaptive filtering.
- Coefficients of some series approximation curve or analytical (as equation) representation.
- Other modeling parameters (for GARH and e.t.c. models).

Beside the modeling parameters clustering, the **residuals after subtraction the series model with from the original series** can be also taken as feature or as input for feature extraction.

> *Note*
>
> As a rule by the term clustering, we mean clustering of time series segments. However, in some specific cases the full series or each point values can be clustered. *For instance, in point-wise anomaly detection we will use point values clustering*.

Example of time series segments and its PCA dimension reduction

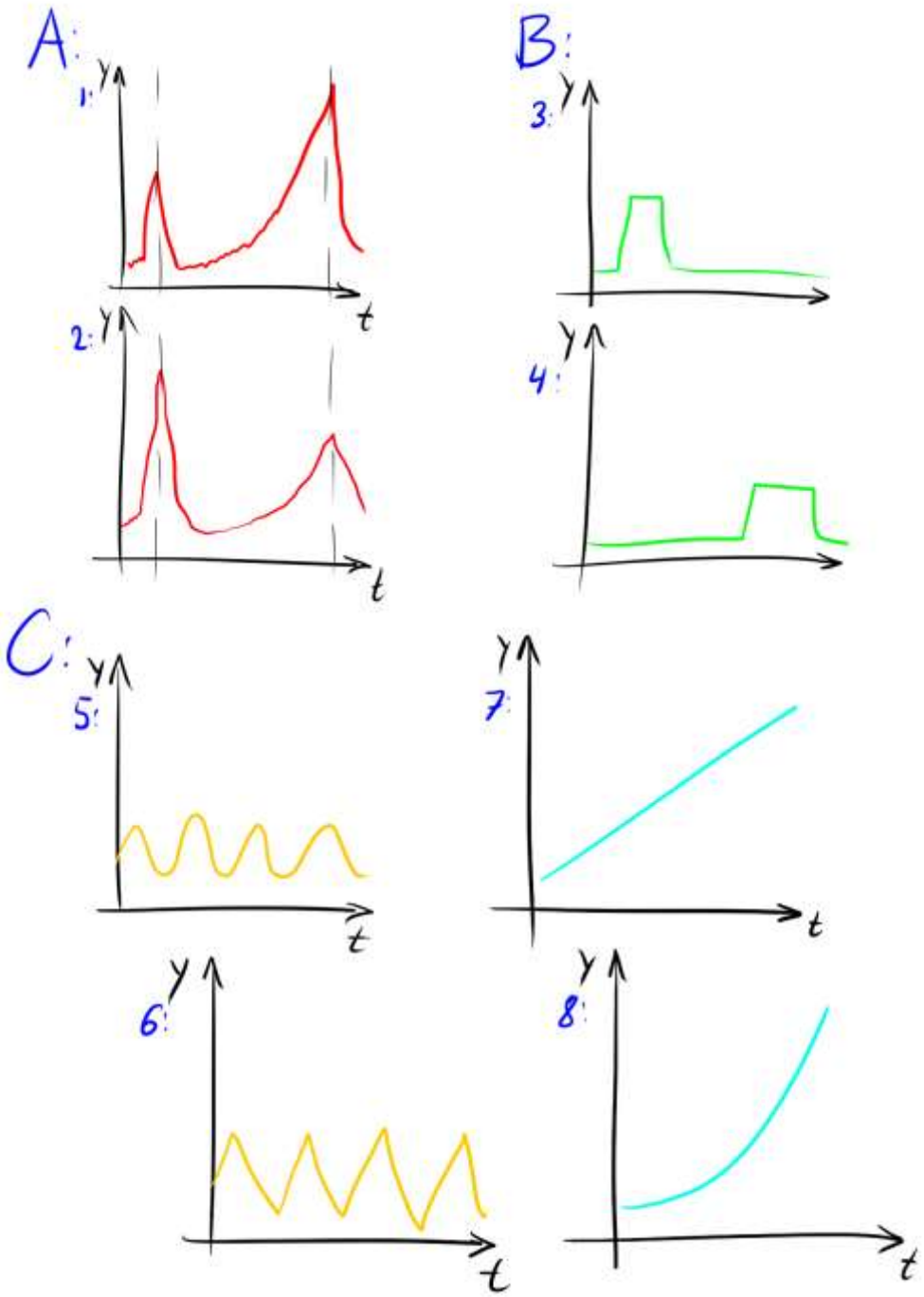


Actually there can be **several types of clustering tasks** for time series.

- **Similarity in time behavior** (like for 1 and 2 on the picture below).
- **Similarity in shape** without time relation (like for 3 and 4 on the picture below).
- **Similarity in pattern** or structure behavior (like for 5 and 6 by seasonality period and 7 and 8 by trend (growing trend without) on the picture below).

Choose of the clustering type depends on the choose of the data representation or algorithm of similarity searching.

e.g. in spectrum domain similarity in shape behavior will be represented in amplitude spectrum similarity. The same for 5 and 6 instances.



1.2.2 Methods of Time Series Clustering

There are exist a plenty methods of clustering, **most popular of them are based on the K-means, and similar ideas.**

K-means

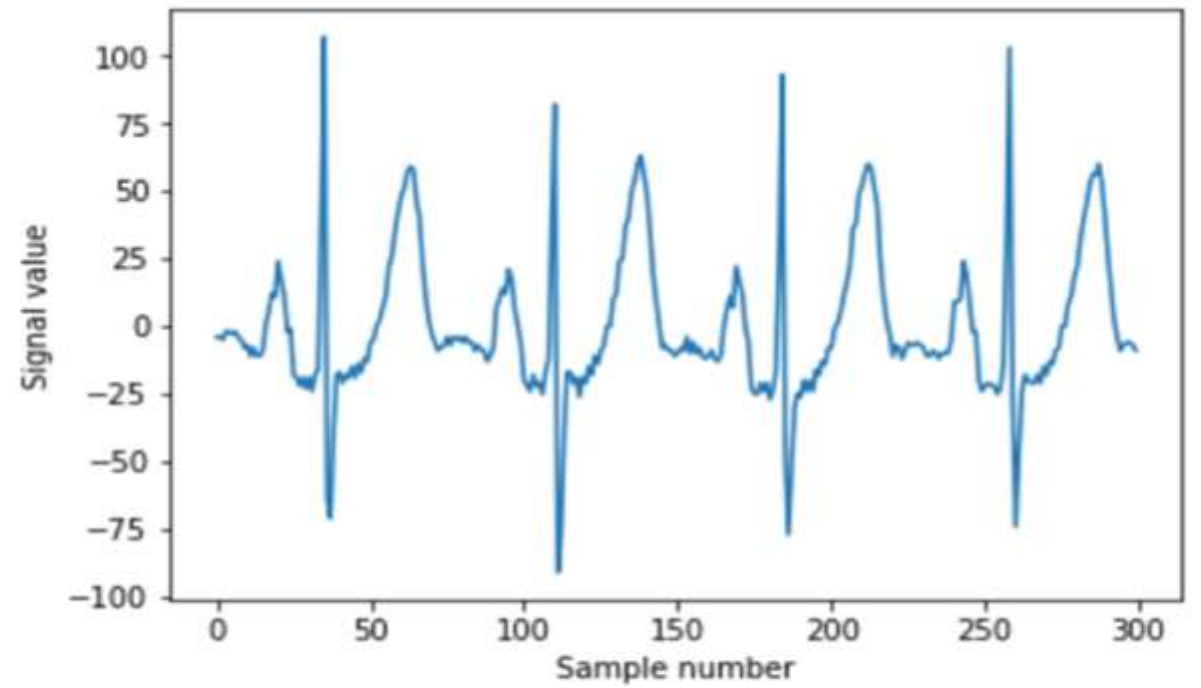
The method is based on the

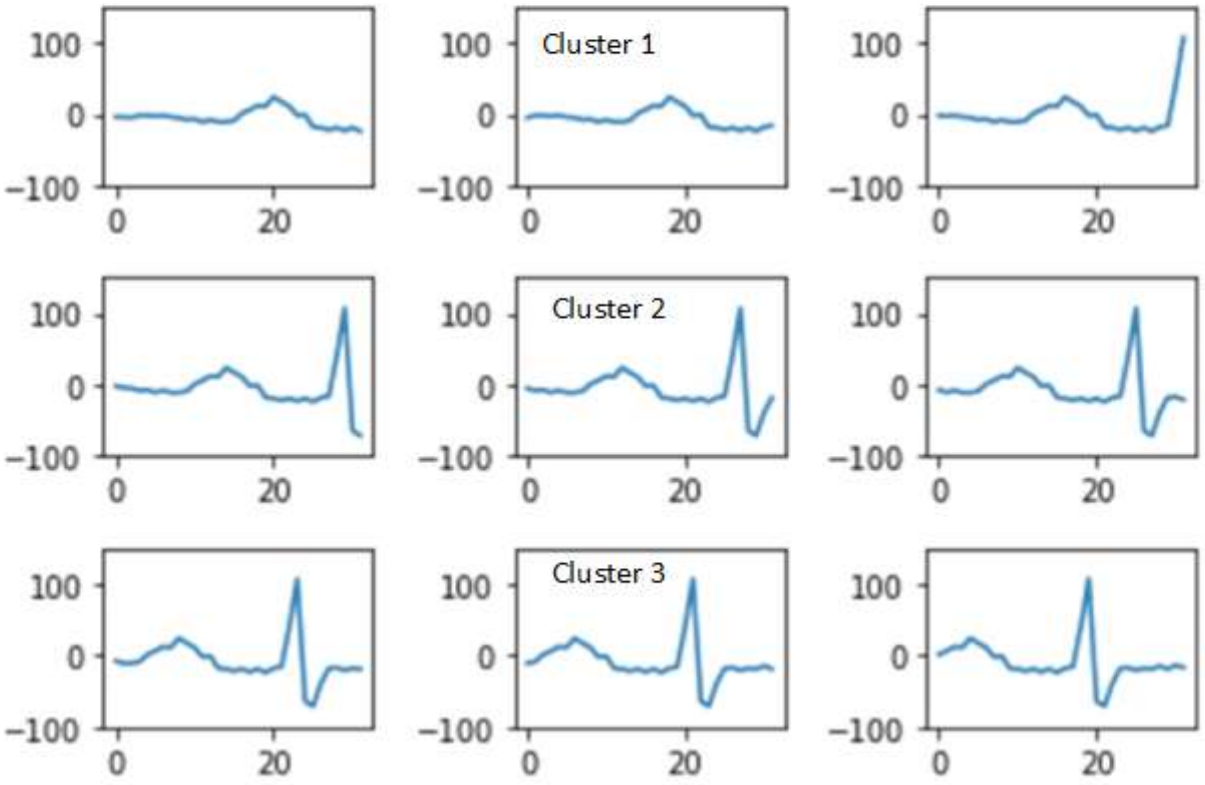
- calculation distance between each pair of points (or patterns, set of points).
- the points (or patterns, set of points) are grouped by searching its centroids (minimization of average distance value in each cluster).

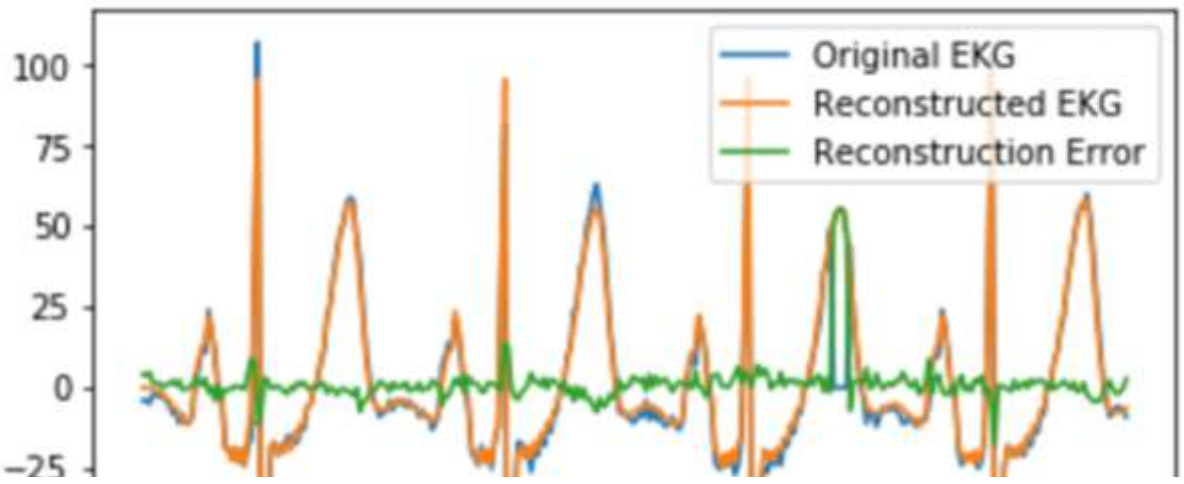
The main drawback of the simple K-means method with Euclidean distance is its instability to various changing in the patterns from one segment to another, e.g. pattern phase delay, shifts, speed of changing or level difference between patterns.

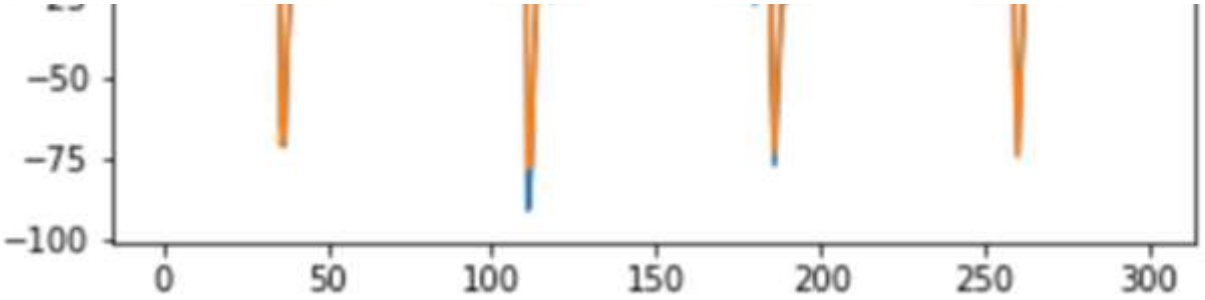
Thus other distances need to be used.

Example of clustering


Signal value
Sample number


Cluster 1
Cluster 2
Cluster 3


Original EKG
Reconstructed EKG
Reconstruction Error



The most important advantage of k-means method is its simplicity and speed.

So it can be applied to large data sets.

However, the algorithm may not produce the same result in each run and cannot handle the outlier.

Beside the distances itself it can be added additional coefficients for some parameters accounting.

for instance, **Computational complexity**,

For instance on the picture below we want to have green and red segments in one cluster,

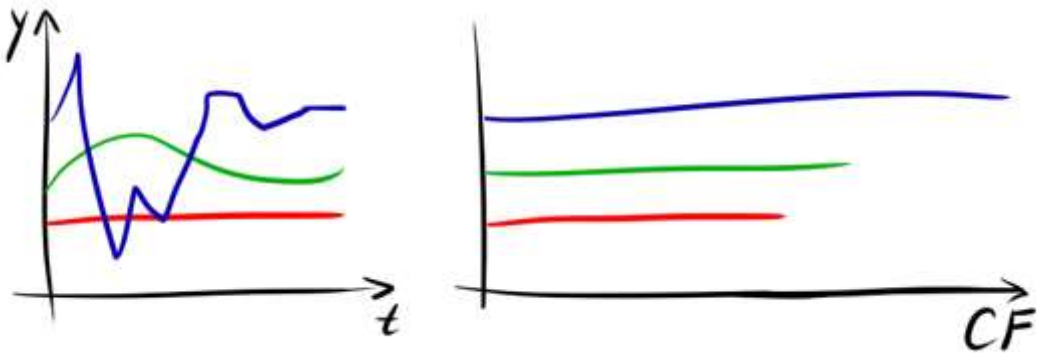
For that we can introduce

Computational complexity as:

$$d_{CF}(x, y) = d(x, y) \cdot CF$$
$$CF = \frac{\max(CE(x), CE(y))}{\min(CE(x), CE(y))}$$
$$CE(x) = \sum_{i=0}^{M_x} \sqrt{(x_i - x_{i-1})^2 + (n_i - n_{i-1})^2}$$

where CE is the complexity, for uniform time steps

$CE(x) = \sum_{i=0}^{M_x} \sqrt{(x_i - x_{i-1})^2}$



Density-based spatial clustering of applications with noise (DBSCAN).

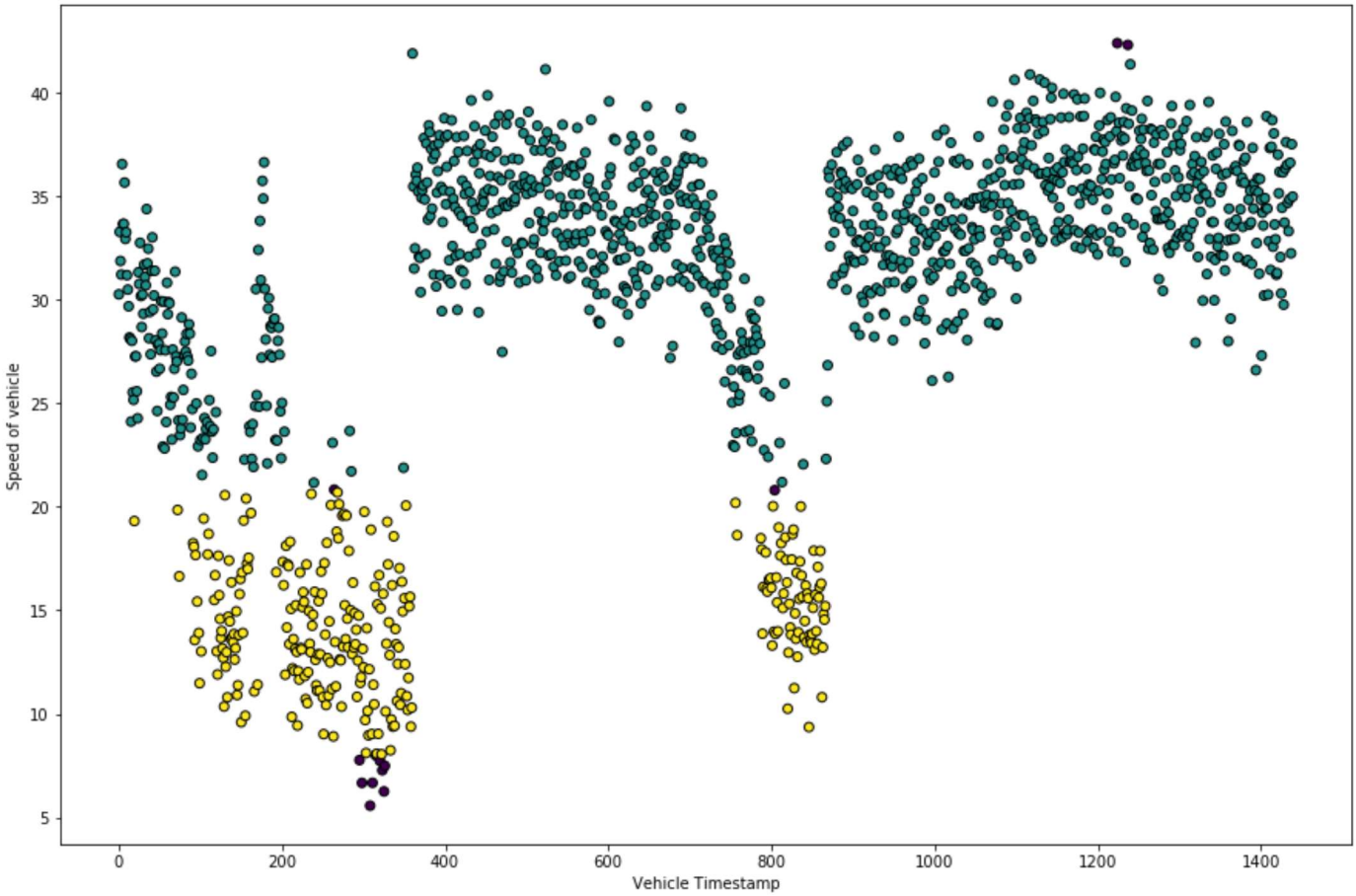
The method is the modified K-means method, but instead of the original ones **in DBSCAN we require a**

- **maximum distance (radius of cluster)**
- **and a minimum number of points in the cluster.**

Thus, **centroids are the points in which the distance less than**

maximum, and the number of instances is maximum (not less than the pre-set minimum number).

Example of time series clustering made by DBSCAN



Other methods of time series clustering:

- Self-Organized Map (Cohenin network),
- Expected Maximization (EM),
- Hierarchical,
- c-mean (fuzzy-logic clustering),

- other more complex.

1.3 Distances for Time Series Comparison

The choose of the correct distance is one of the most complex task in time series as in clustering, as in classification adn so on.

The **basic distances** are:

- Euclidian Distance,
- Generalized Euclidian Distance and Mahalanobis Distance,
- Median Distance, and other robust distance types,
- Mean Absolute Distance and other Minkowski Distances,
- Mean Absolute Percentage Distance,
- Correlation Distance (or Cosine Distance, Pearson Coefficient),
- Probability Distribution Based Distances,
- Dynamic Time Warping Distance,
- more complex techniques, based on the feature extraction and feature transformation.

Note

The main problem with the time series distance is the influence of some segments fluctuations on the resulted value. Thus, if it is necessary we need to test some complex distances or test some pipelines of feature extraction and feature transformation with distances.

Euclidian Distance:

$$d = \frac{1}{M}\sqrt{\sum_{i=0}^{M-1}(x_i - y_i)^2},$$

where d is the distance between two series segments x and y both with length M.

Mean Absolute Distance and other Minkowski Distances,

$$d = \frac{1}{M}\sum_{i=0}^{M-1}|x_i - y_i|,$$
$$d_p = \frac{1}{M}\sum_{i=0}^{M-1}|x_i - y_i|^p,$$

Generalized Euclidian Distance and **Mahalanobis Distance**,

$$d = \frac{1}{M}\sqrt{\sum_{i=0}^{M-1} w_i(x_i - y_i)^2}, d_m = \frac{1}{M}\sqrt{\sum_{i=0}^{M-1}\frac{(x_i - y_i)^2}{\sigma(x_i)\sigma(y_i)}},$$

where w_i is the weight coefficient; σ_i is the covariation between x_i and y_i the case $w_i = \sigma_i$ - is the Mahalanobis Distance.

Mean Absolute Percentage Distance,

$$d = \frac{1}{M}\sum_{i=0}^{M-1} |\frac{x_i - y_i}{x_i}|,$$

Median Distance:

$$d = median(\{x_i - y_i\}_{i=0}^{M-1}),$$

where $median$ is the central value of sorted samples $\{x_i - y_i\}$ of size M.

More robust distance then Euclidian Distance and Mean Absolute Distance.

Correlation Distance (or Cosine Distance, Pearson Coefficient),

$$\text{corcof} = d_{cos} = \frac{\sum_{i=0}^{M-1} x_i y_i}{\sqrt{\sum_{i=0}^{M-1} x_i^2 \sum_{i=0}^{M-1} y_i^2}},$$

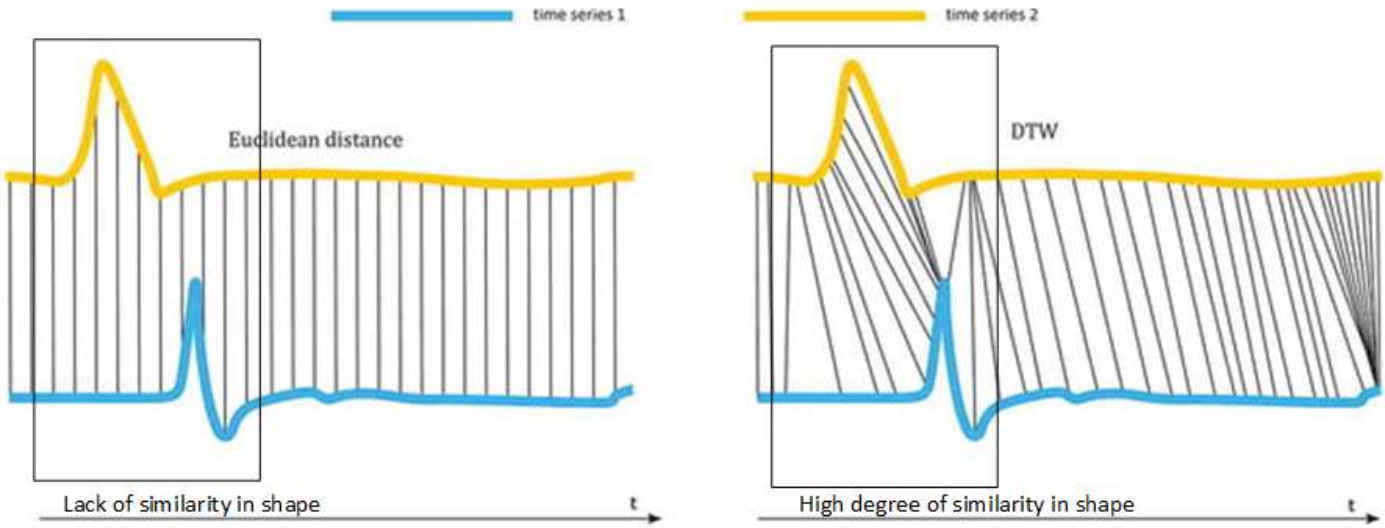
Probability Distribution Based Distances,

There are a plenty of methods for calculation, for instance Kullback-Leibler distance

$$d_{KL} = \sum_{i=0}^{M-1} x_i \log x_i / y_i$$

Dynamic Time Wrapping (DTW) Distance,

Non-linear algorithm, based on the searching of maximum similarity between points independently of its index position.



The **DTW algorithm** has the follows steps:

1. Calculate the distance between the first point in the first series segment and every point in the second series.

Select the minimum of the calculated values and store it (this is the "time warp" stage).

2. Move to the second point and repeat stage 1.

   Move step by step along points and repeat stage 1 till all points are exhausted.

3. Calculate and Select the minimum of distances between the first point in the second series segment and every point in the first series.

4. Move step by step along points in the second segment and repeat stage 3 till all points are exhausted.

5. Sum all the stored minimum distances.

The **DTW algorithm** can be formally described as

$$D(i, j) = 0$$

$$D(i, j) = dist(x_i, y_j) + min\{D(i-1, j), D(i, j-1), D(i-1, j-1)\}$$

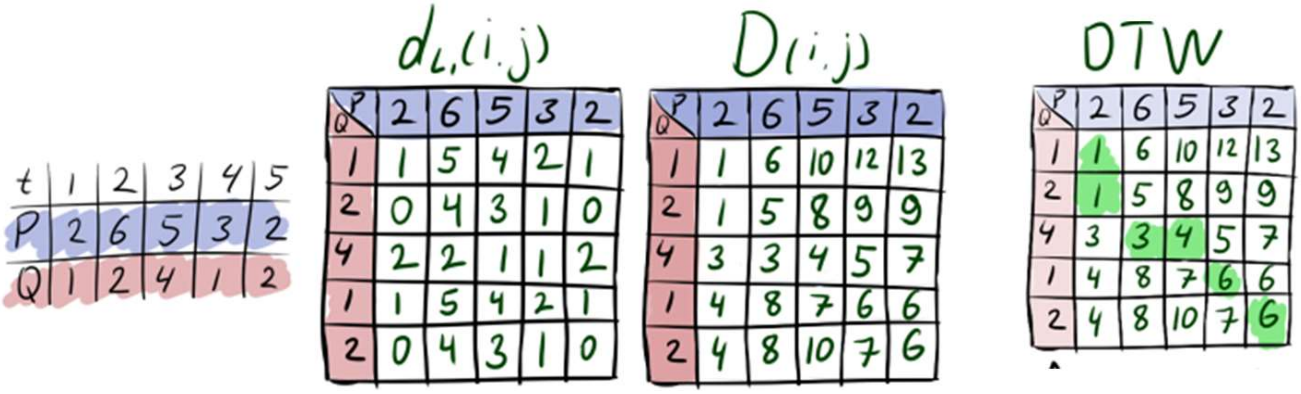
$$d_{DWT} = min_i \frac{\sum_{j=0}^{K} D(i, j)}{K}$$

where

- D is the element of virtual matrix of time wrapping with elements $D(i, j)$ and size $M_x \times M_y$;
- $dist$ is the distance function (for instance, Euclidian distance, or MAE);

- x and y are the series segments with length M_x and M_y correspondingly. Thus here length of segments can be non-equal;
- K is the length of virtual trajectory build form right bottom of the matrix D ($D(M_x, M_y)$) to the up-left point ($D(0, 0)$) such that its sum is minimal).

Example of DTW calculation



Dynamic Time Wrapping (DTW),features

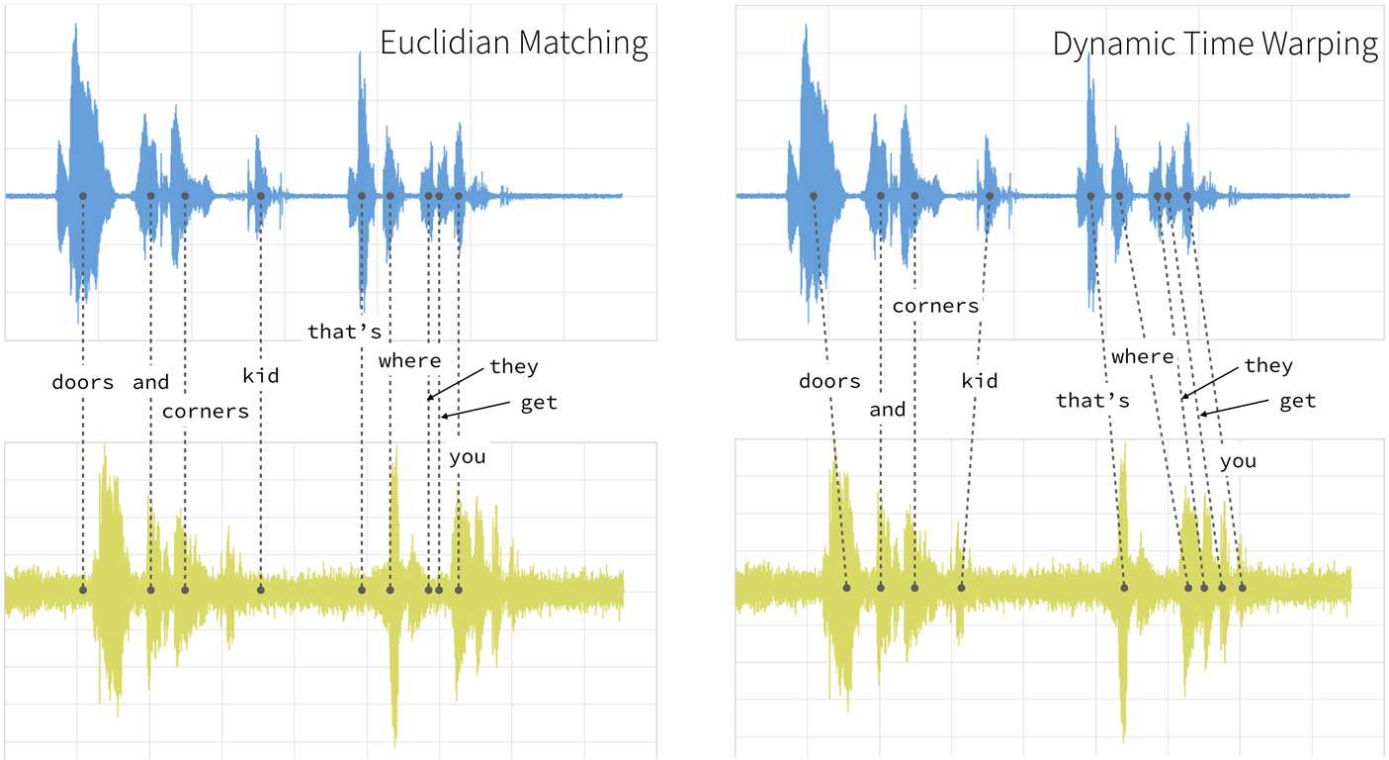
- Non-linear algorithm, based on the searching of maximum similarity between points independently of its index position.
- The main **advantage of the DTW** distance is the independence (invariation) to the small shifts or other slight changes in the segments (like its squeezing and stretching and other variations).

In other words DTW method try to build a matrix of mapping (or transformation) of one segment to another, and find the best (minimum cost distance between them).

- The main **drawback of the DTW** is the high complexity and implicity of similarity search. Thus in some cases the DTW can show similarity where it should not be.
- The other draw back is high computational complexity.

  There are exist a lot of simplified modification of DTW distance.

Example of DTW for speech clustering.

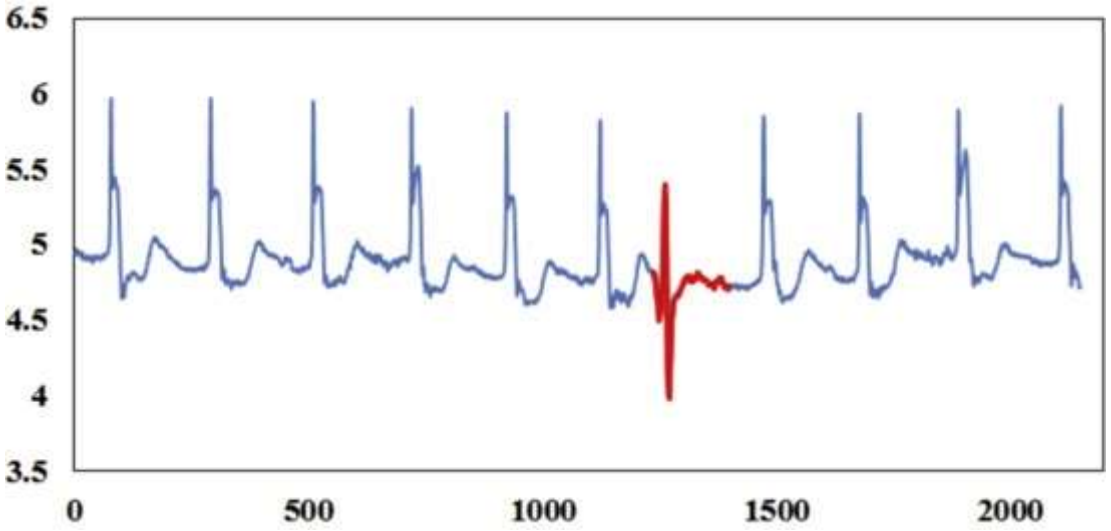


1.4 Anomaly detection

1.4.1 Task of Anomaly detection

The outlier or anomaly detection problem for time series is usually formulated as finding sharp and short-time changing of data values relative to some standard or usual series values.

> Thus anomaly series parts (or segments) must be significantly distinct from rest of the data in statistical meaning.



Form the definition an anomaly (or outlier) is an observation with at least one variable having an unusual value.

- A **univariate outlier** is an observation with a variable that has an unusual value.

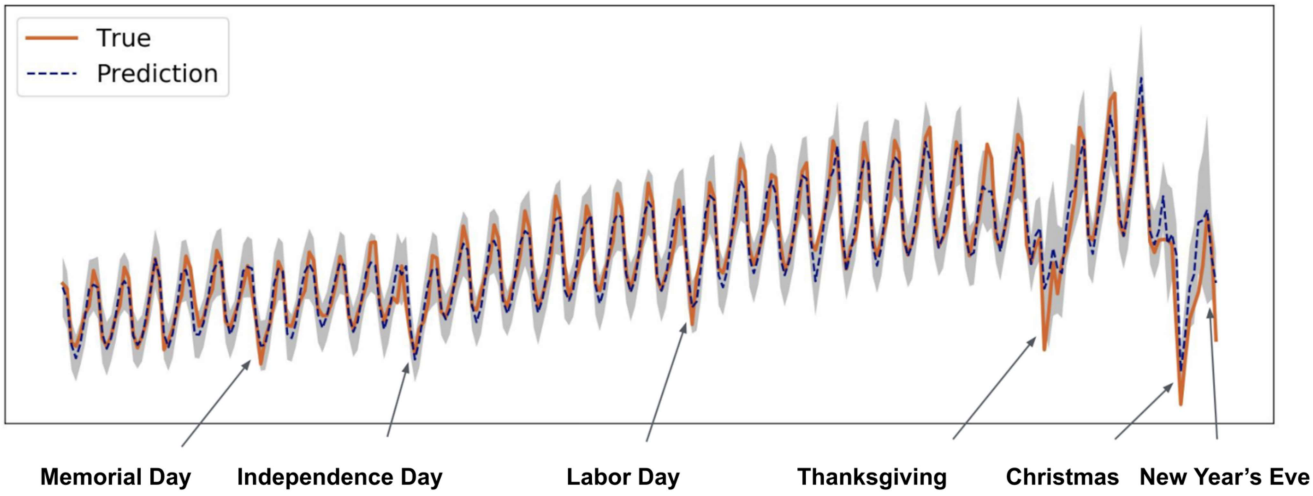
- A **multivariate outlier** is an observation with at least two variables having unusual values.

In literature, different terms are used that have the same or similar meaning to Anomaly Detection.

Types of Anomalies in Time Series Data

- Irregular event detection;
- Novelty detection;
- Deviations discovery;
- outliers detection;
- Change Point Detection;
- Fault detection (fault diagnostic);
- Fraud detection;
- Misuse detection.

Example of the anomaly interpritation in sales prediction.

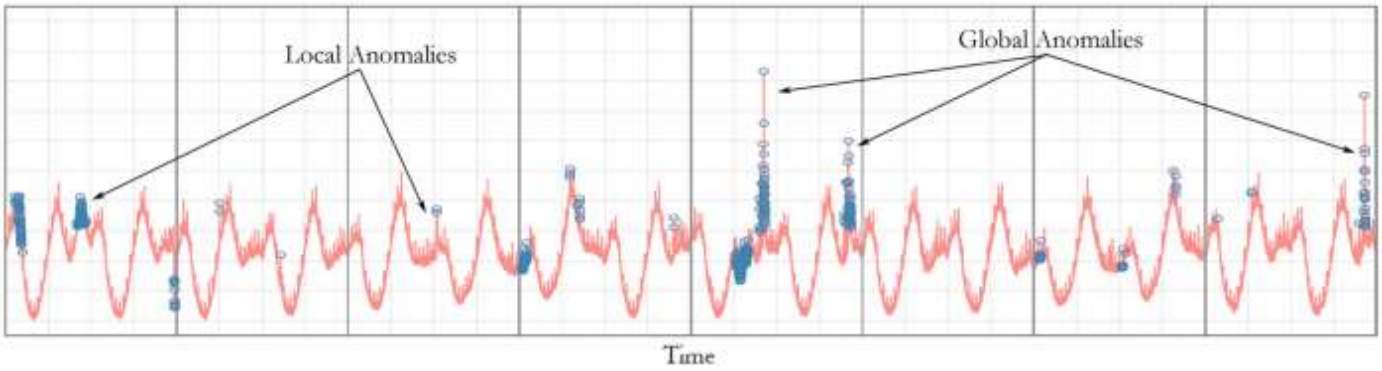


There can be distinguished the local and global anomalies.

- The **local anomaly** assume that its value does not overcome general values range.
- The **global anomaly** assume that its value significantly overcome general values range.

  Example of local and global point-like anomalies.

  Local anomaly mean that its value does not exceeds average envelope value.



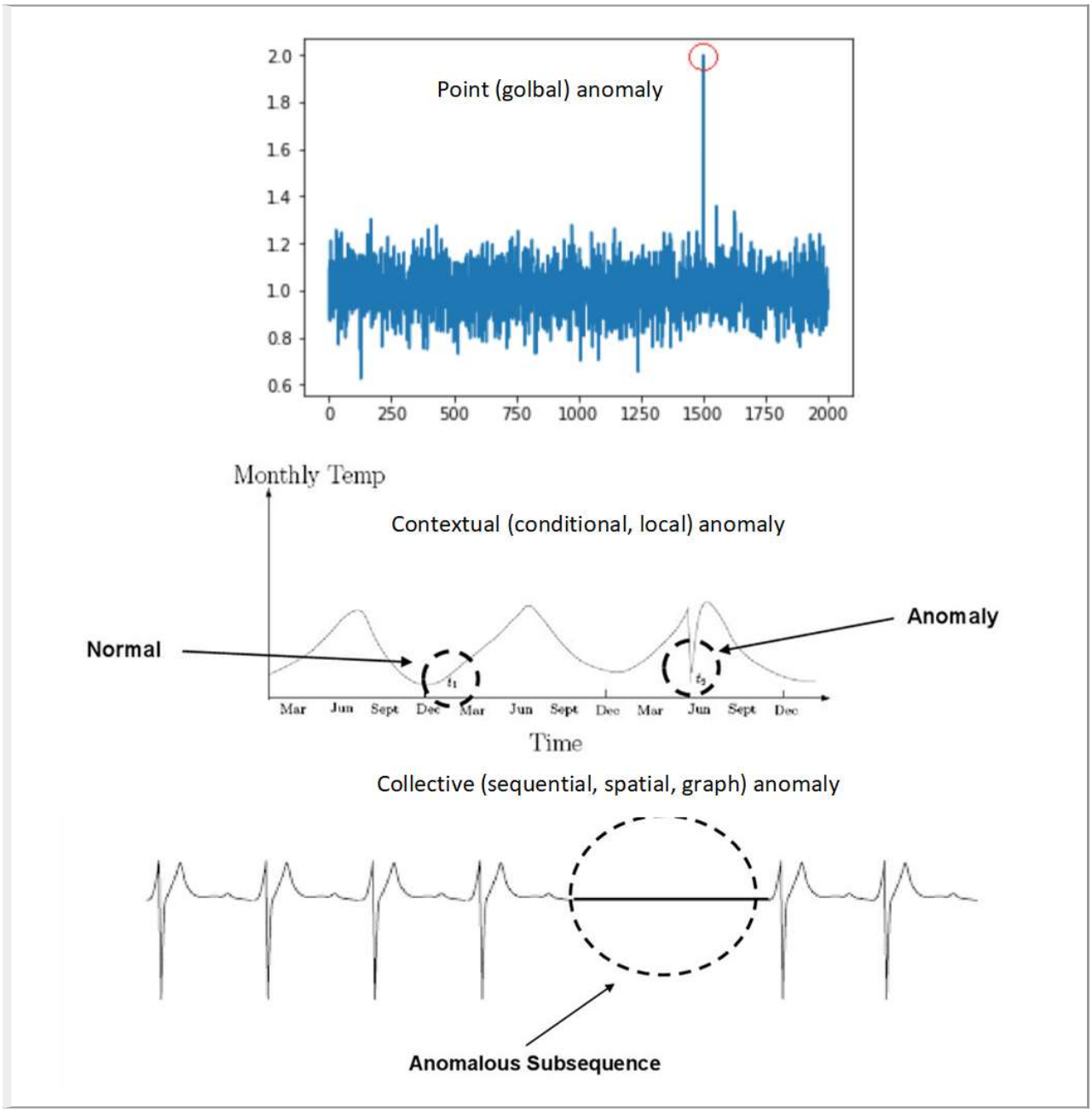
The following **types of anomalies** (by its behavior) can be distinguished.

- Sudden **point-like** metric (or sample) value **spike**, absence or collapse.
- Sudden and short-time **change in variance**.
- Sudden and short-time **level shift**.
- Sudden and short-time **envelope change**.
- Sudden and **short-time lack of seasonality or changing** of its behavior.
- **Combination** of the previous cases.

Note

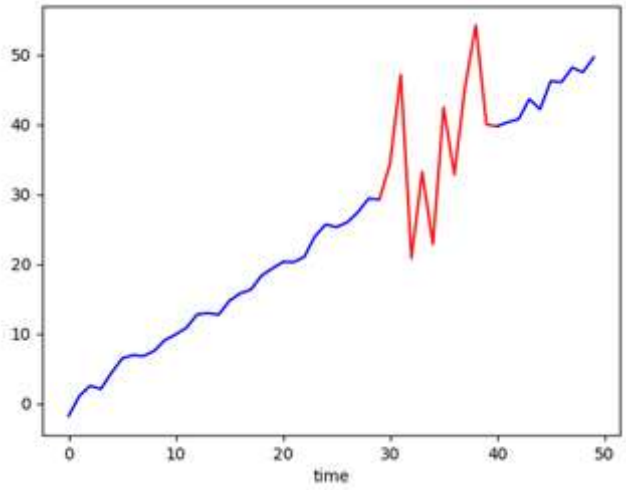
The relative long anomaly behavior are some times named contextual anomaly or collective anomaly.

Contextual anomalies are data instances that are, in human terms, “not being themselves”.

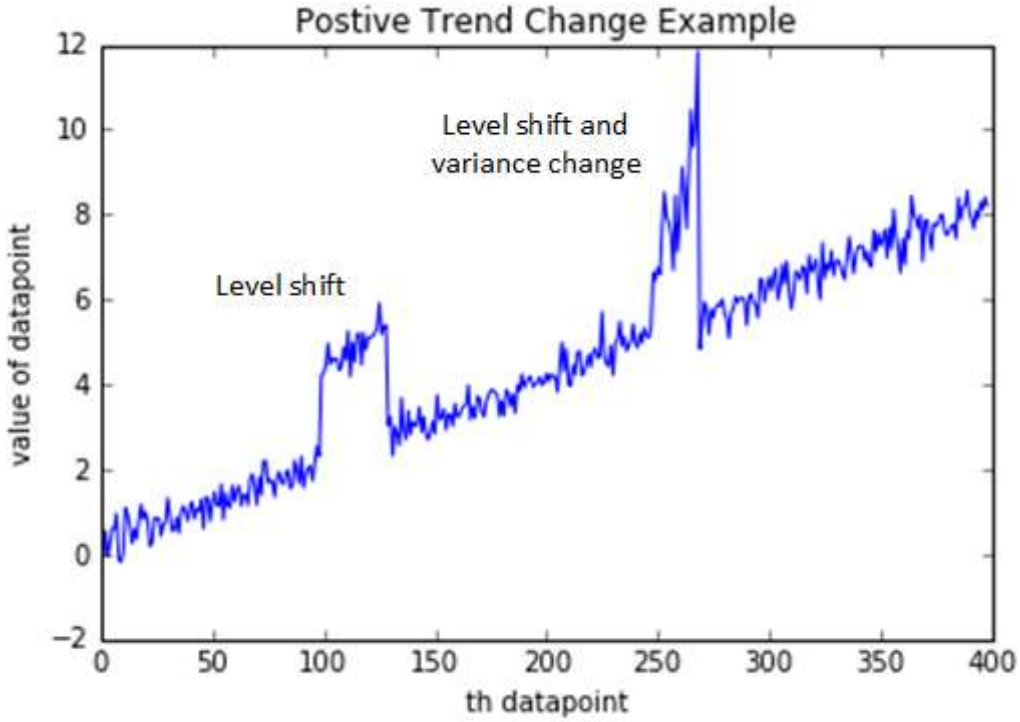


1.4.2 Examples of Anomaly Detection

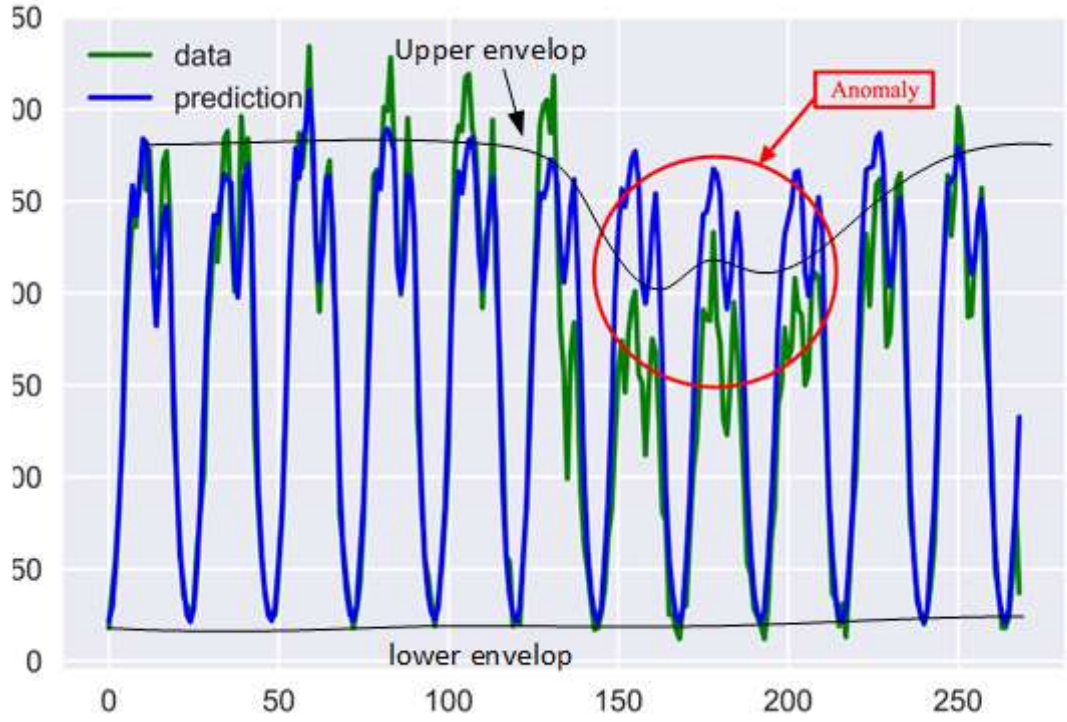
- Example of variance change anomaly.



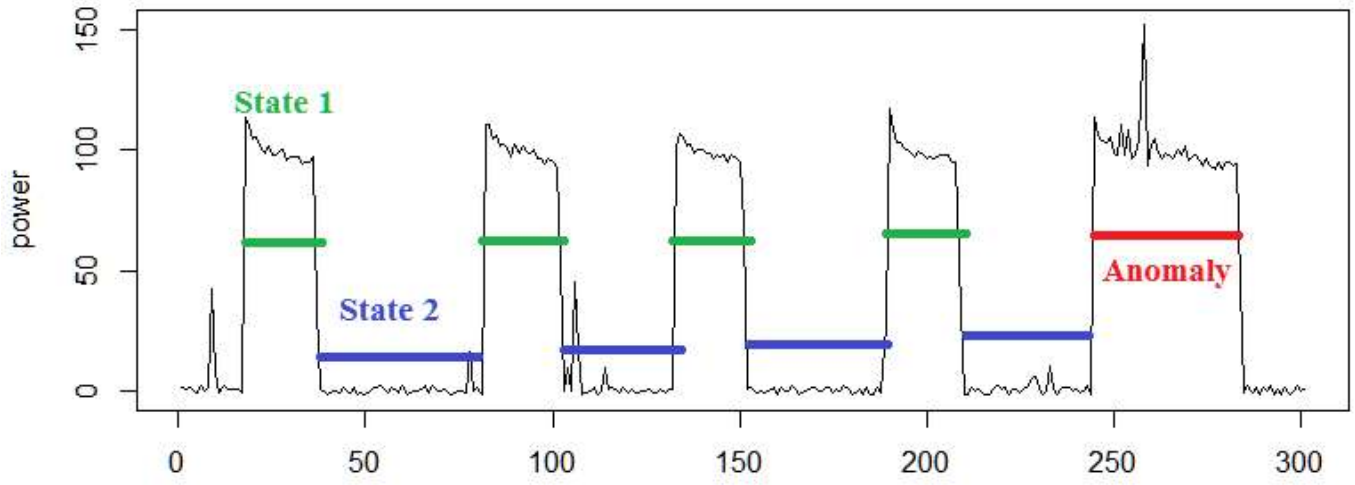
- Example of level shift anomaly.



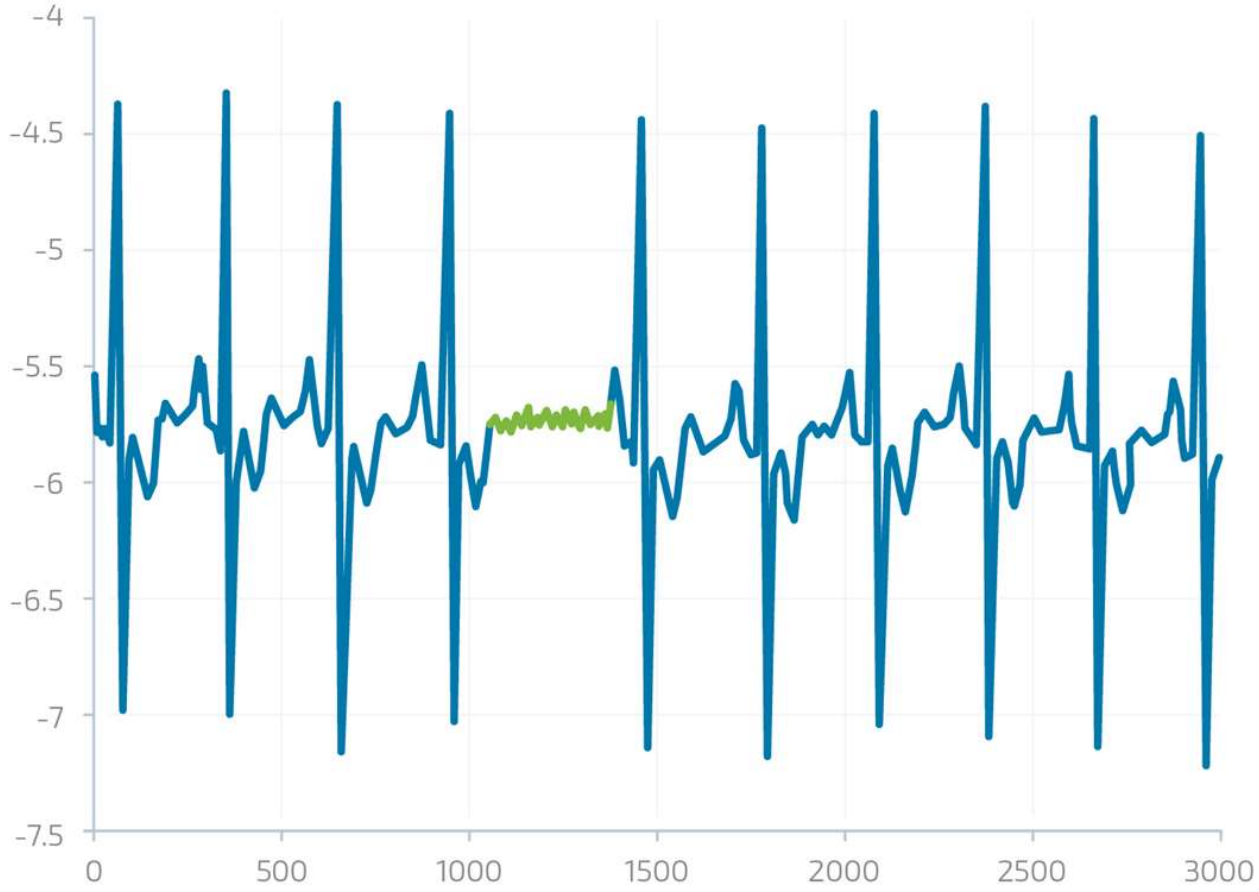
- Example of envelope change anomaly.



- Example of seasonality behavior anomaly.



- Example of sudden lack of seasonality and level shift anomaly.



It could be distinguish a several approaches for anomaly detection:

1. **Anomaly detection based on error of the prediction** of the time series (and deviation of it).
2. **Anomaly detection based on unusual shapes (patterns, behavior)** of the time series.

In each approach we have to introduce some metric for anomalies detection (in classic statistic terms deviation) .

Such metric can be, for instance:

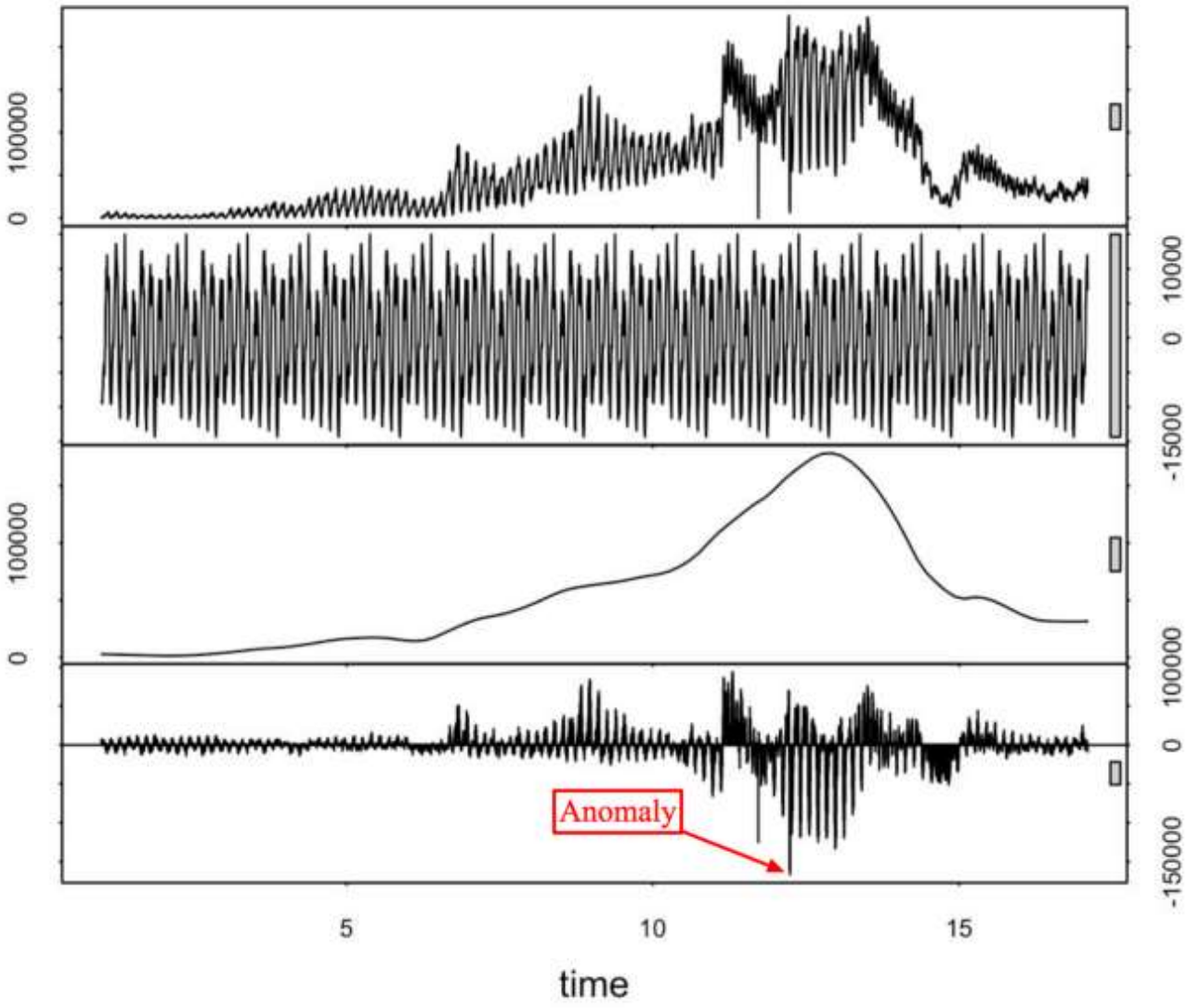
- Euclidean distance,
- MAPE (Mean Absolute Percentage Error),
- Some statistic information metric like an entropy or KL divergence.

In All cases the estimated deviation - i.e. error value is proportional to the anomaly score.

If the anomaly score is above a some set threshold, it is marked as an anomaly.

Note the anomaly search can be carried out as for full time series as for results of it decomposition.

Example of anomaly search in the residuals of decomposition.



1.4.3 Methods of Anomaly Detection

The Anomaly detection can be carry out using:

- **supervised Anomaly detection methods**

  - the series with two classes, labeled as normal and anomaly,

- **semi-supervised Anomaly detection methods**

  - the series has one class, labeled as normal and many unlabeled data,

- **unsupervised Anomaly detection methods**

  - the unlabeled series, but you have the rule for choose which normal data must satisfy.

In many cases as base-line for anomaly detection methods comparison the **Interquartile Range (IQR)** can be applied.

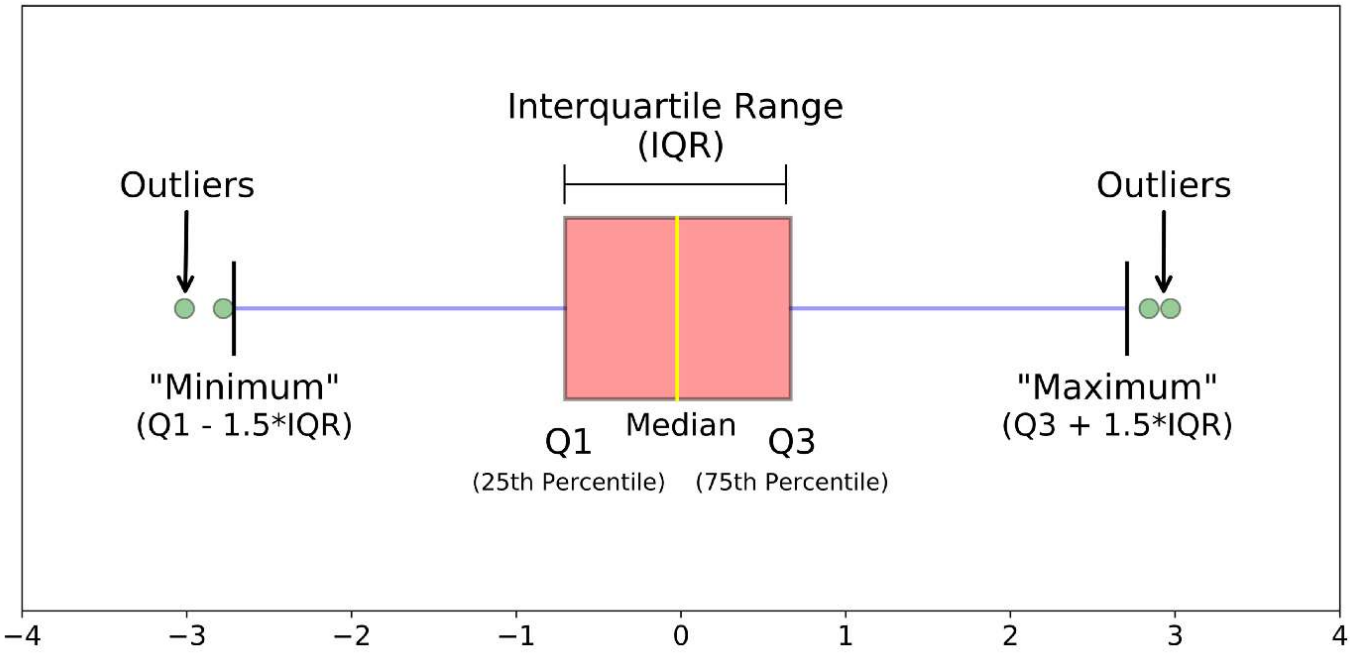
- The method is representation of data distribution as 4-quantiles.
- The outliers can be considered as all values above 75% or below 25% quartiles.
- As a rule IQR are graphically represented as the **Box Plot**.

If you arrange data from small to large, the mid-point is called the median. The median splits data into two halves. The mid-points of each of these halves is called a quartile.

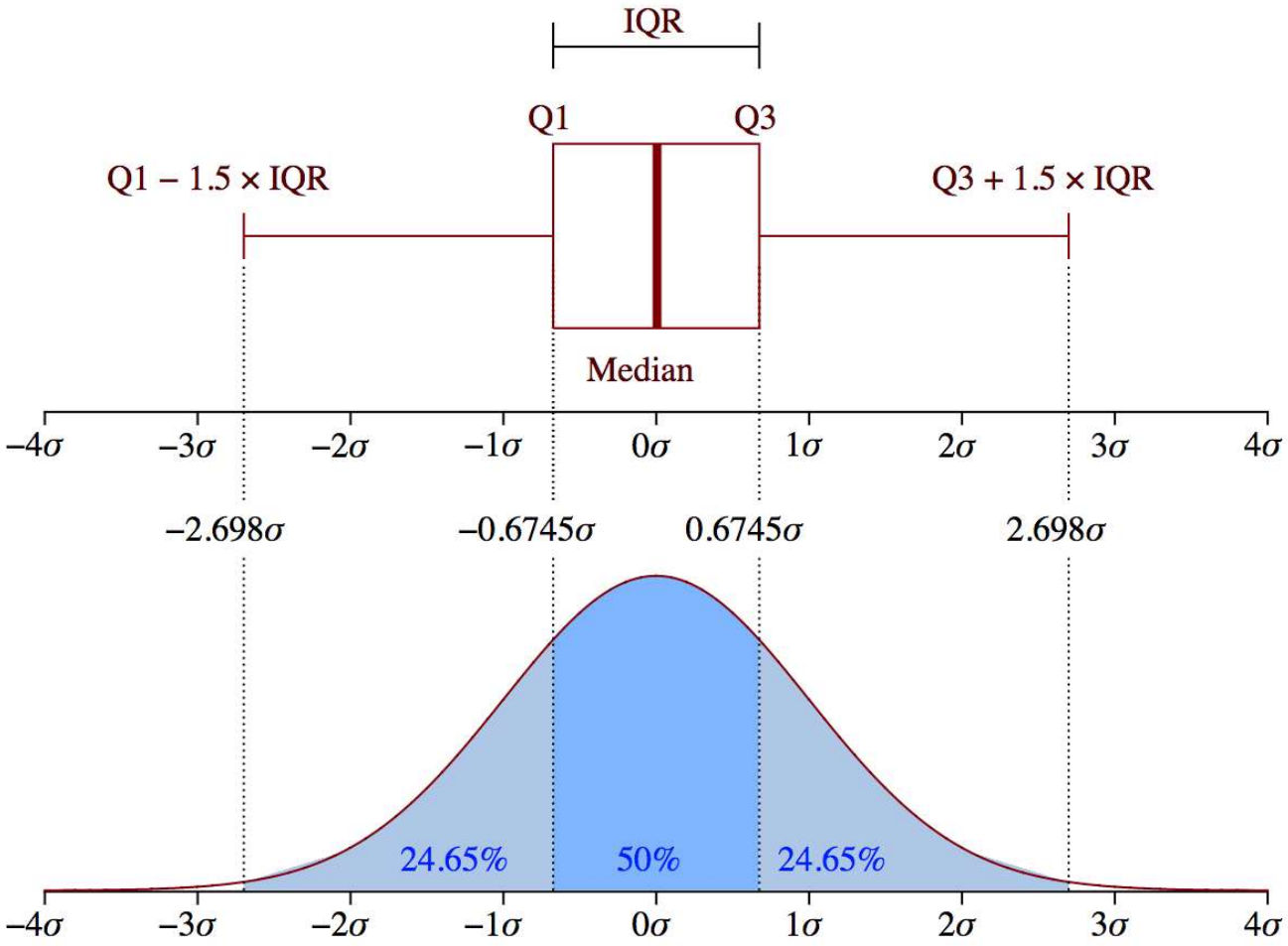
In other words, you can split data into 3 quartiles — 1st, 2nd and 3rd (the 2nd quartile has a name for it— the median). So the Interquartile Range is the distance between 1st and 3rd quartiles. The theory behind anomaly detection using IQR is that, if a data point is too far from the 1st and 3rd quartile, it probably is an outlier.

IQR can be used standalone for outlier detection, but boxplots below use the same algorithmic theory and are probably more intuitive than IQR.

In the boxplot below, the length of the box is IQR, and the minimum and maximum values are represented by the whiskers. The whiskers are generally extended into 1.5*IQR distance on either side of the box. Therefore, all data points outside these 1.5*IQR values are flagged as outliers.



Comparison between Box Plot and Normal distribution



1.4.3.1 Unsupervised methods

Most of the **unsupervised** anomaly detection methods try to determine anomalies by analyzing the distribution of all error values and use the percentile value threshold.

One widespread approach is to set threshold = $3 \cdot std$ where std is the standard deviation of the considered distribution.

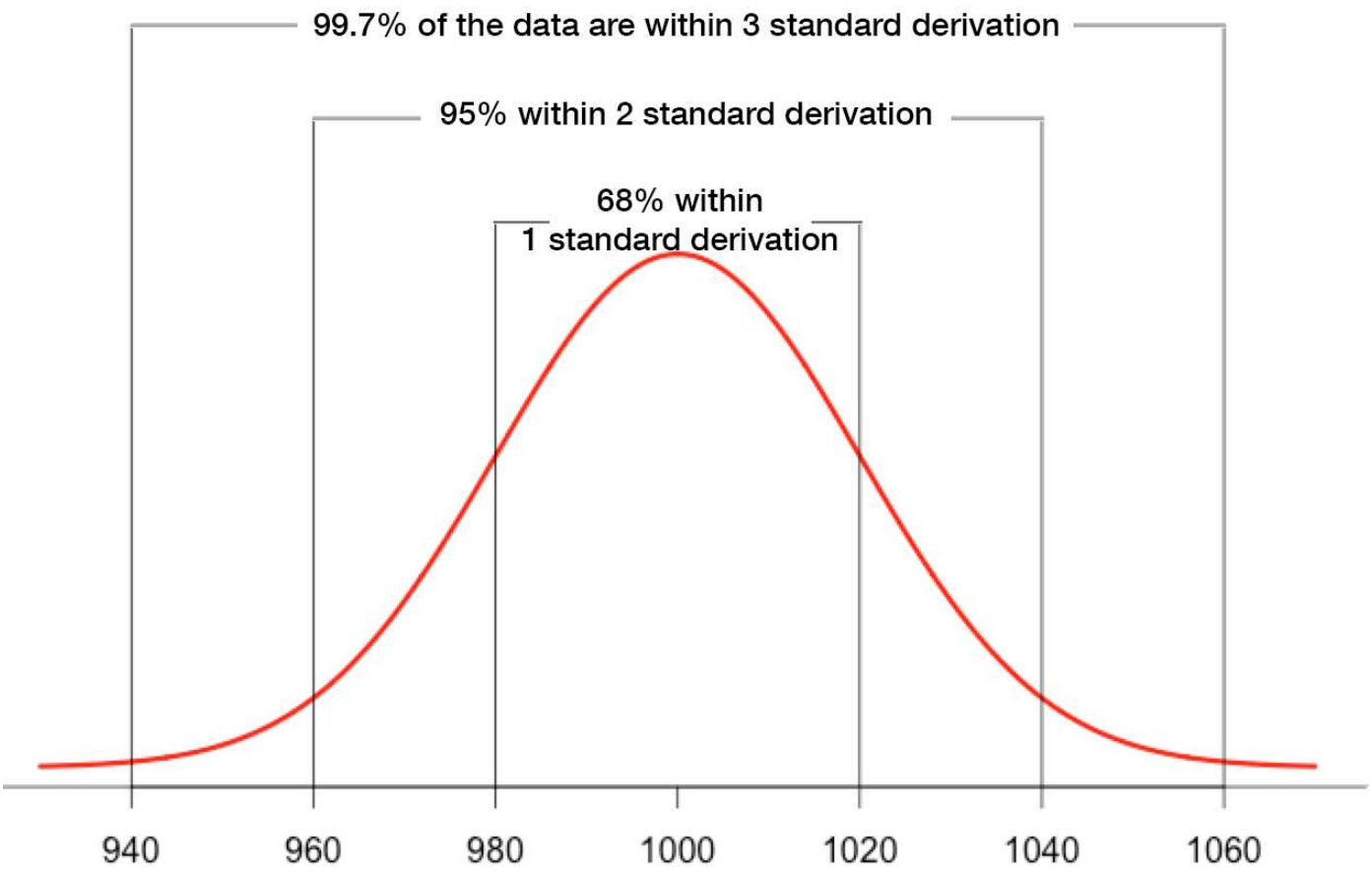
The data can be pre-normalized to estimate its values (and outliers) in relative units

$$z = \frac{y - EV(y)}{std(y)}.$$

all the z values above 2.5 or below -2.5 should be considered as outliers with confidence 99%. Illustration of the **percentile value threshold** as example of

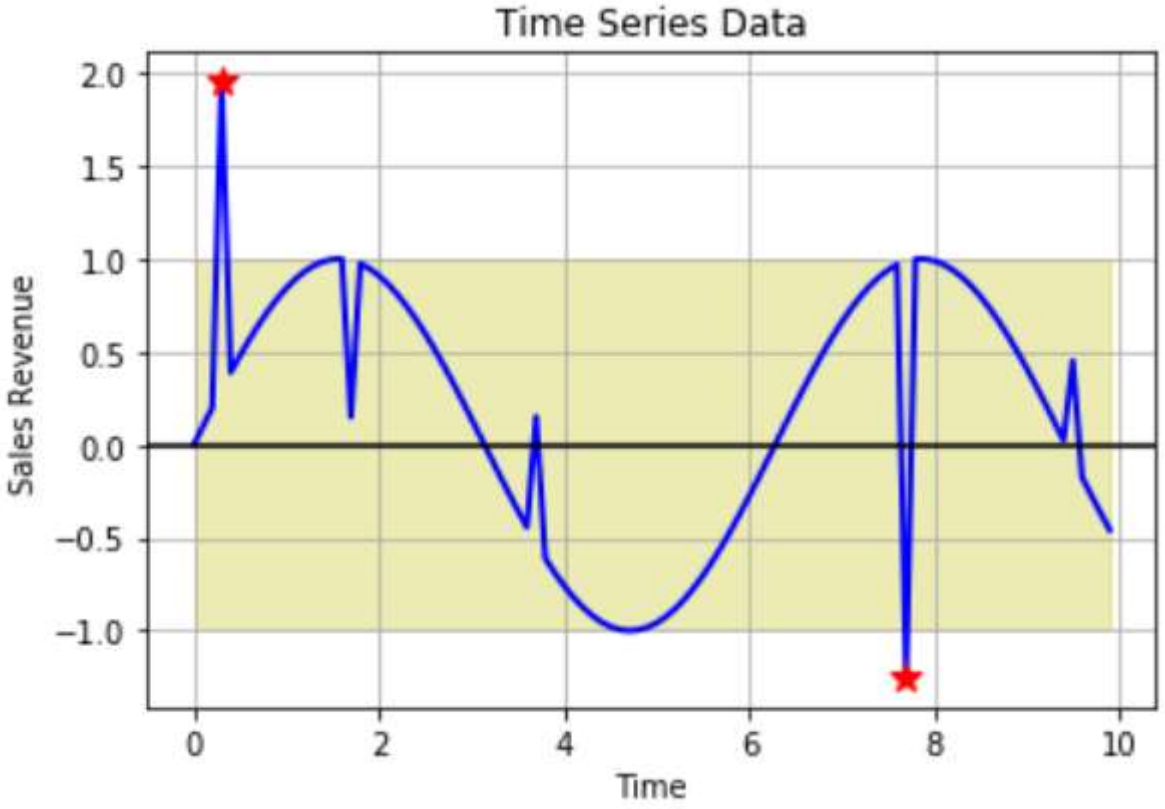
68-95-99.7 rule or **three-sigma rule of thumb**.

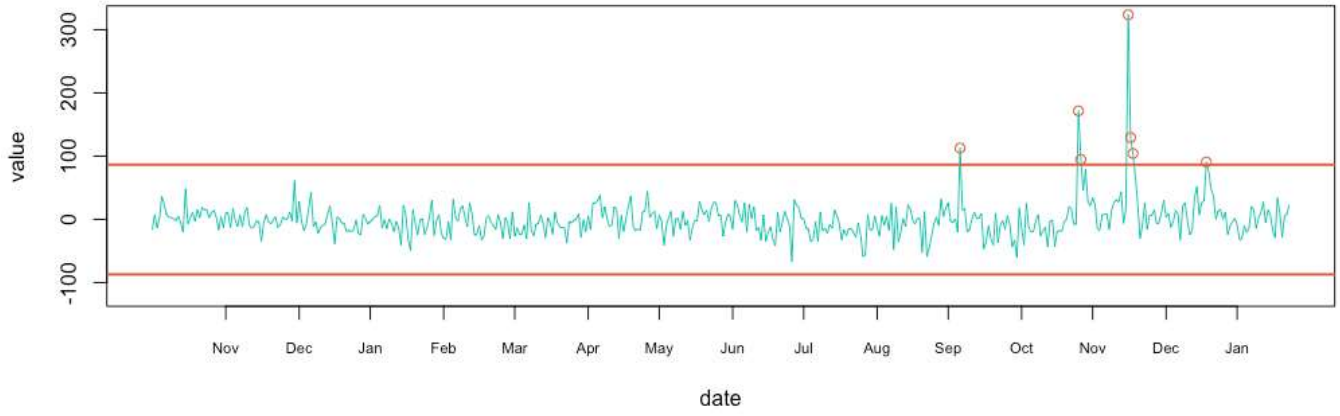
- 68% of all values fall between $[mean - std, mean + std]$;
- 95% of all values fall between $[mean - 2 \cdot std, mean + \cdot std]$;
- 99,7% of all values fall between $[mean - 3 \cdot std, mean + 3 \cdot std]$.

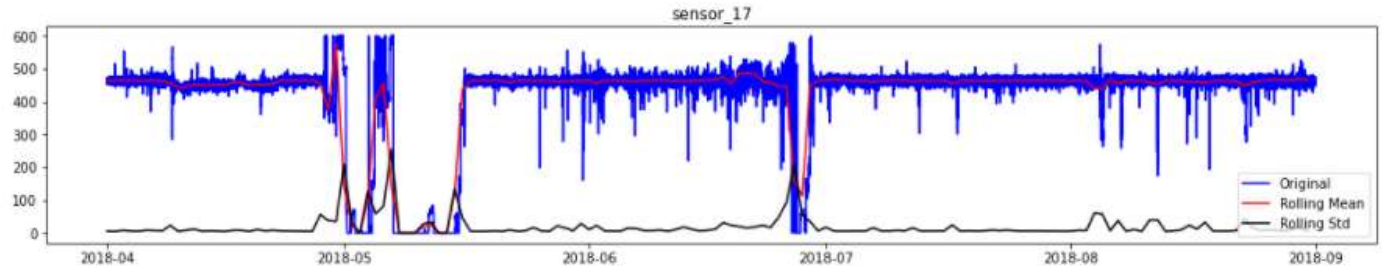


Global threshold method - the most simple technique which suit for global outlier thresholds is the.

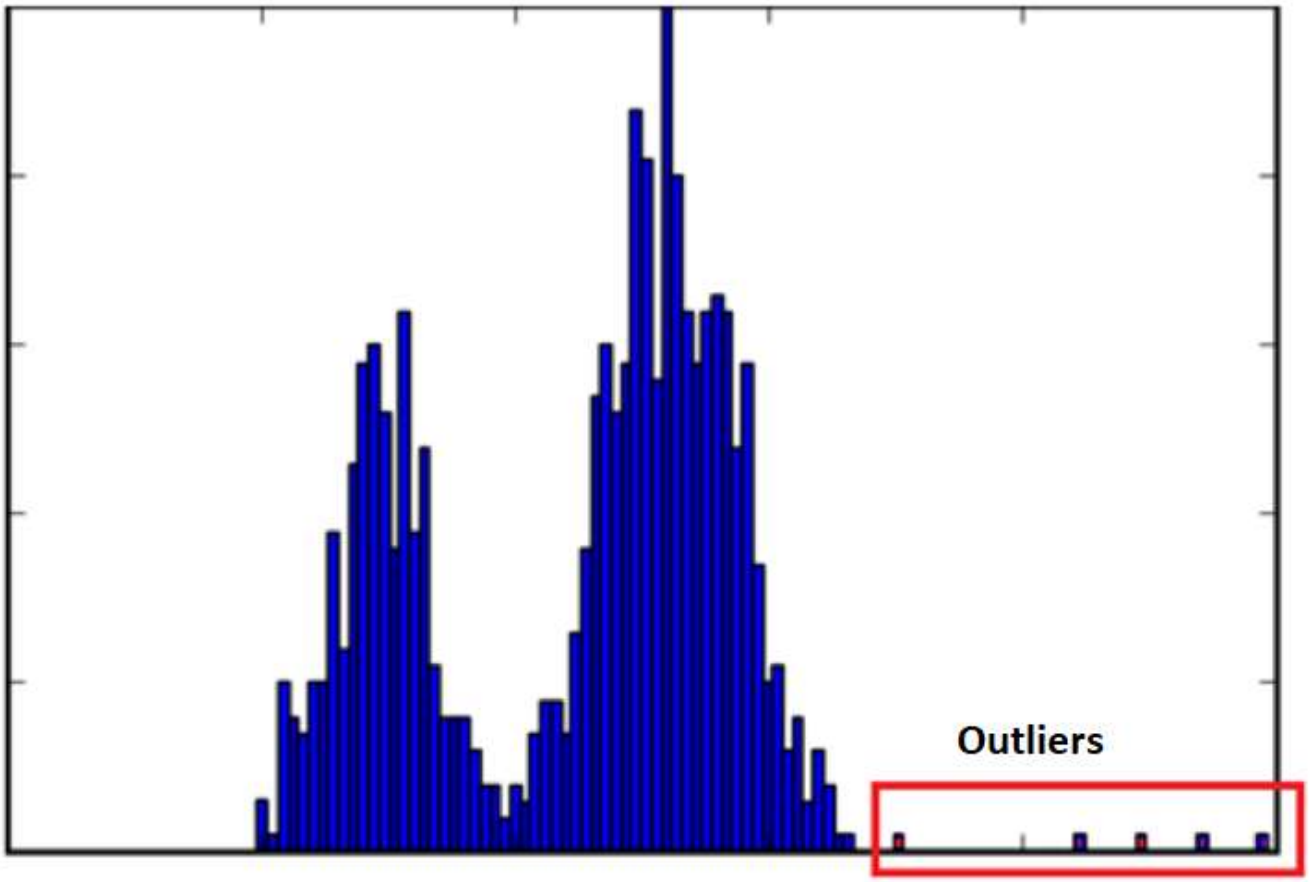
> The searching can be done by calculating statistical values like mean or median moving average of the data and using a standard deviation to come up with a band of statistical values which can define the uppermost bound and the lower most bound and anything falling beyond these ranges can be an anomaly.



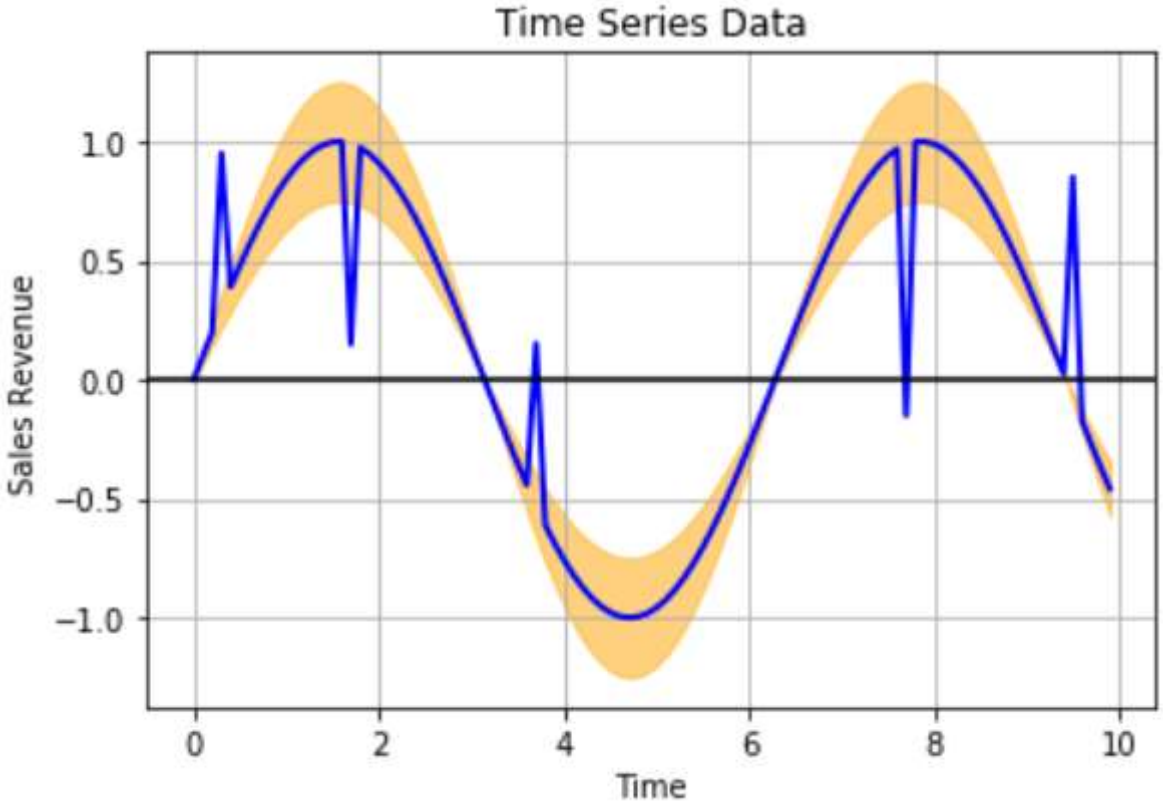


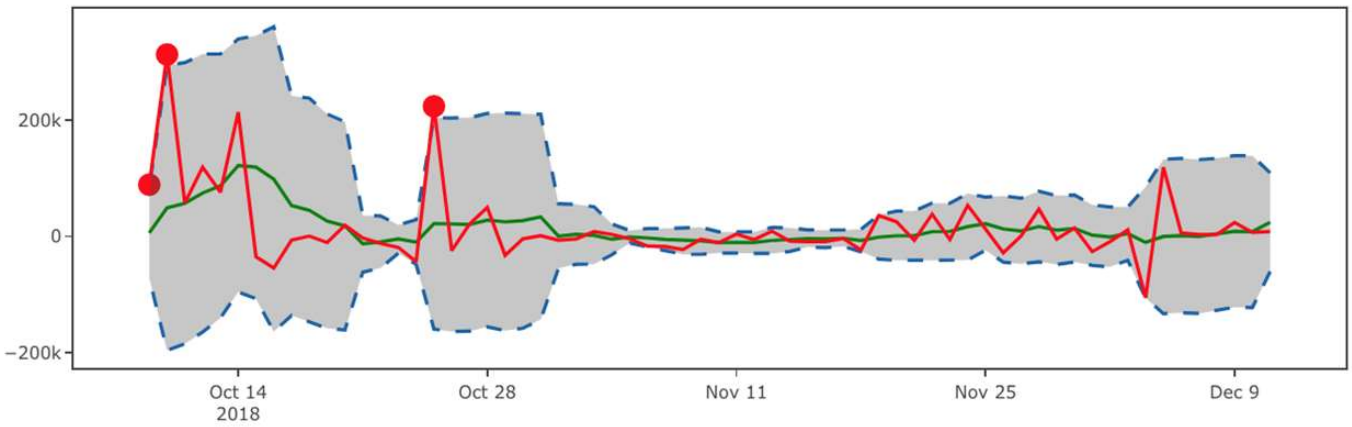


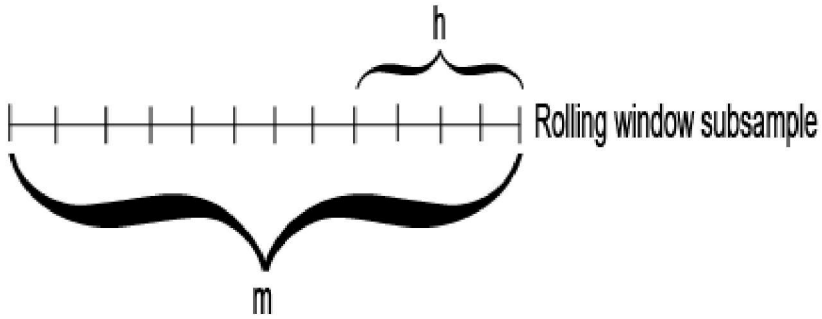
Example of threshold selection based on the distribution estimation.

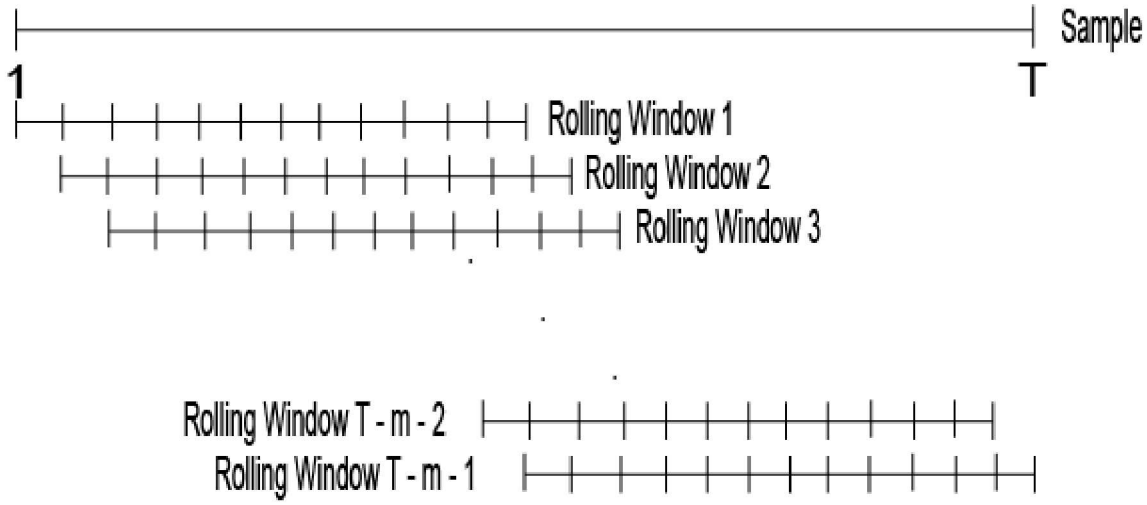


Local threshold method for anomalies searching. Analyze anomalies by relative threshold in some moving window. For instance, relative to the average std or mean.


Time Series Data
Sales Revenue
1.0
0.5
0.0
−0.5
−1.0
0
2
4
6
8
10
Time


200k
0
−200k
Oct 14
2018
Oct 28
Nov 11
Nov 25
Dec 9





Model based unsupervised search

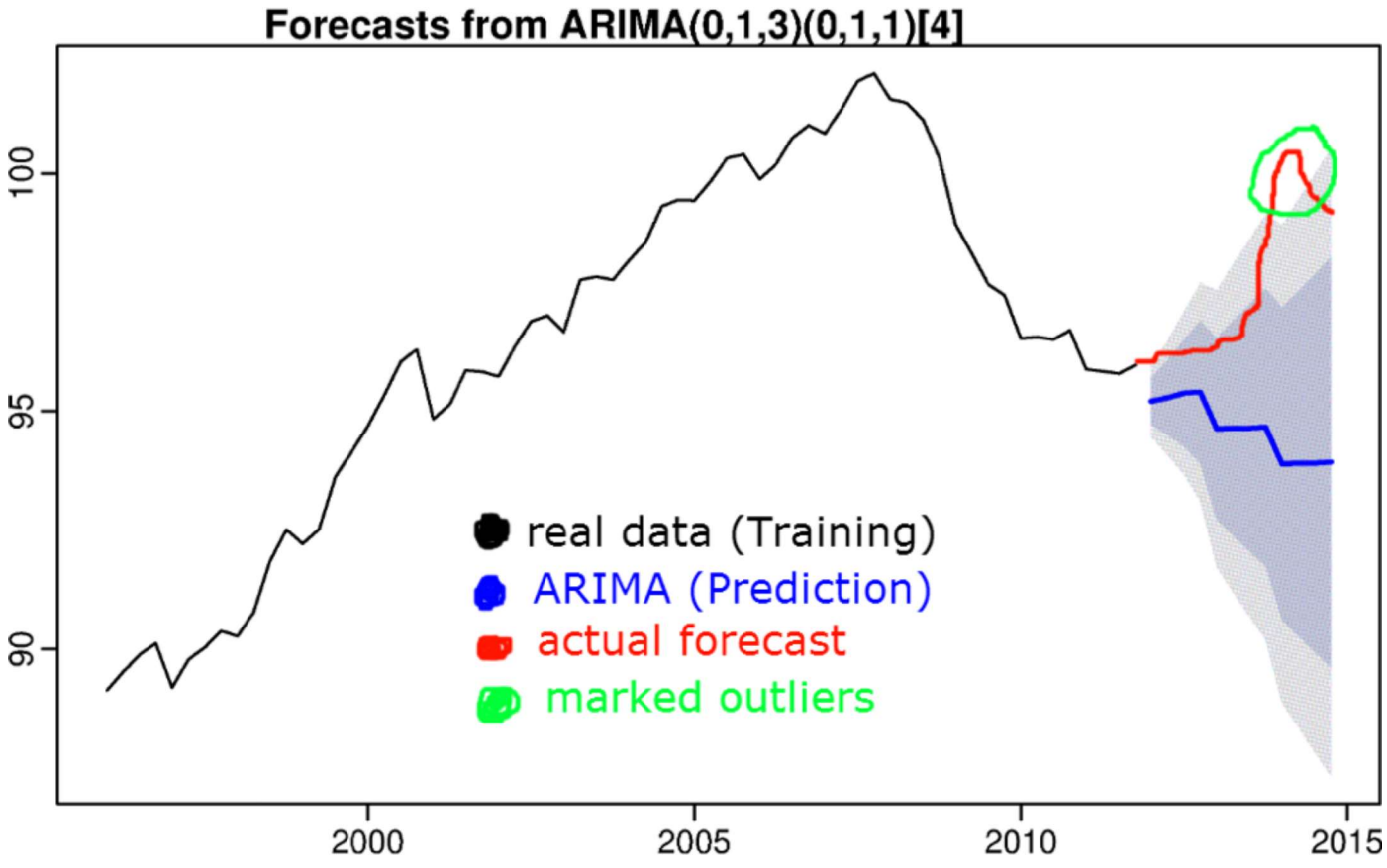
The method is based on the creation of the predictive model and set threshold of the prediction error (for instance MAPE for local anomalies).

- The error threshold here can be interpreted as confident level for approve hypothesis of outliers.
- This approach is suit for both local and global anomalies.
- For building the predictive model, popular time series modeling algorithms like moving average, SARIMA, or any Regression or

Machine Learning and Deep Learning based algorithm like LSTM can also be used effectively.

> *Note*
>
> If model parameters estimation carried out only for "normal" data (with out anomalies) than the task can be considered as semi-supervised.

Model based example of outlier search by prediction.



Clustering based anomaly search

The method is based on the assumption that anomalies will be partied into separate clusters with relatively low amount of elements.

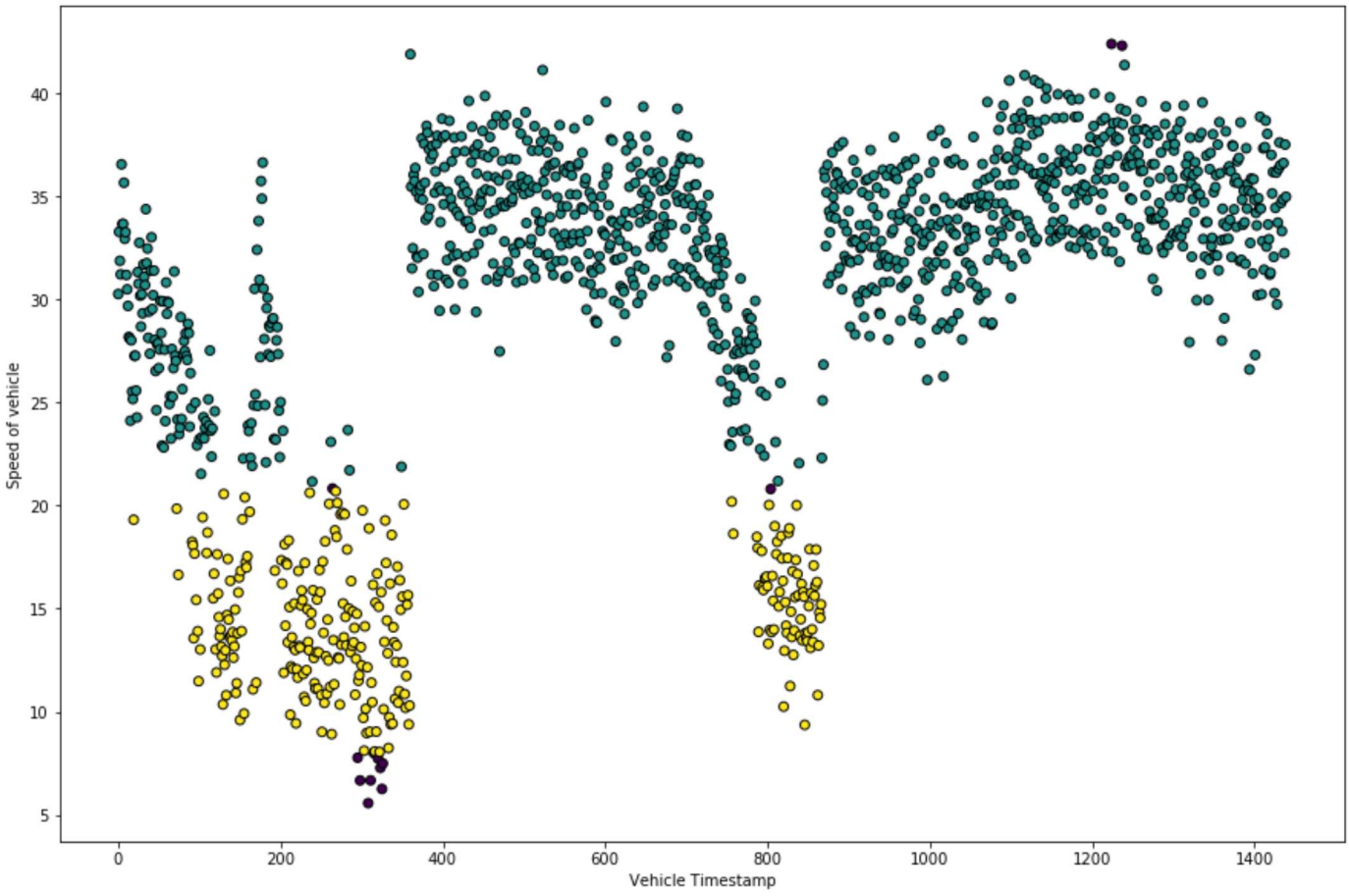
- One common pitfall or bottleneck for clustering algorithms for anomaly detection is defining the number of clusters, which is required by most clustering algorithm as an input.
- The Density Based clustering methods like **DBSCAN** becomes the natural choice due to the lack of number of clusters requirement.
- The main draw back of such clustering if anomaly data points repeats many times in a sparse interval, these might not be mapped as an anomaly. So, in such a case, going with a rolling window based clustering.

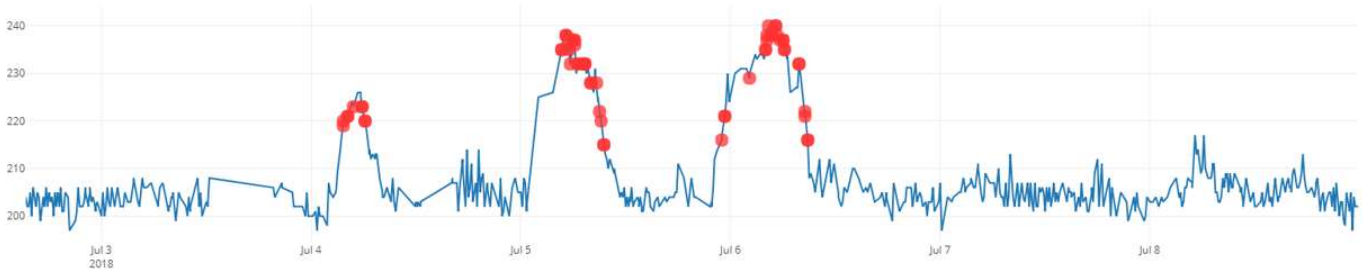
Specificity of **DBSCAN in Anomaly detection**:

- Any point that has at least minimum samples points within set distance of it will form a cluster.
- Any point within maximum cluster distance of a central point, but does not have minimum samples points within distance of itself is called a borderline point and does not form its own cluster.

- Any point that is not found to be a central point or a borderline point is called a noise point or outlier and is not assigned to any cluster.

Thus, such points does not contain at least minimum samples points within maximum distance from it or is not within set distance of any determined central point.





Local Outlier Factor

The density based k-means-like method for unsupervised anomaly detection.

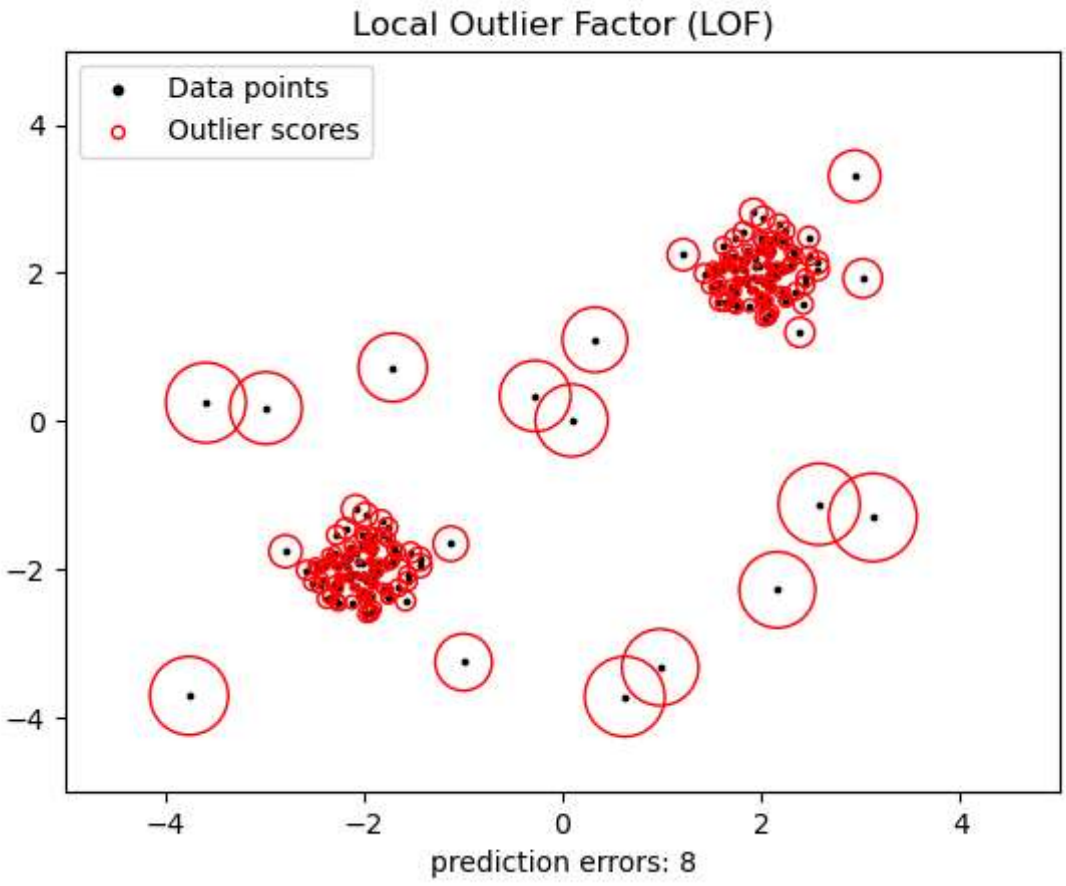
- The method is based on the calculation of the inverse of the average distance between each point and its k neighbors.
- The average of normalized density if local outlier factor.
- Higher LOF value for any point indicate a greater anomaly level.

The LOF method is for searching points that are outliers with respect to their local neighborhood (not gobal).

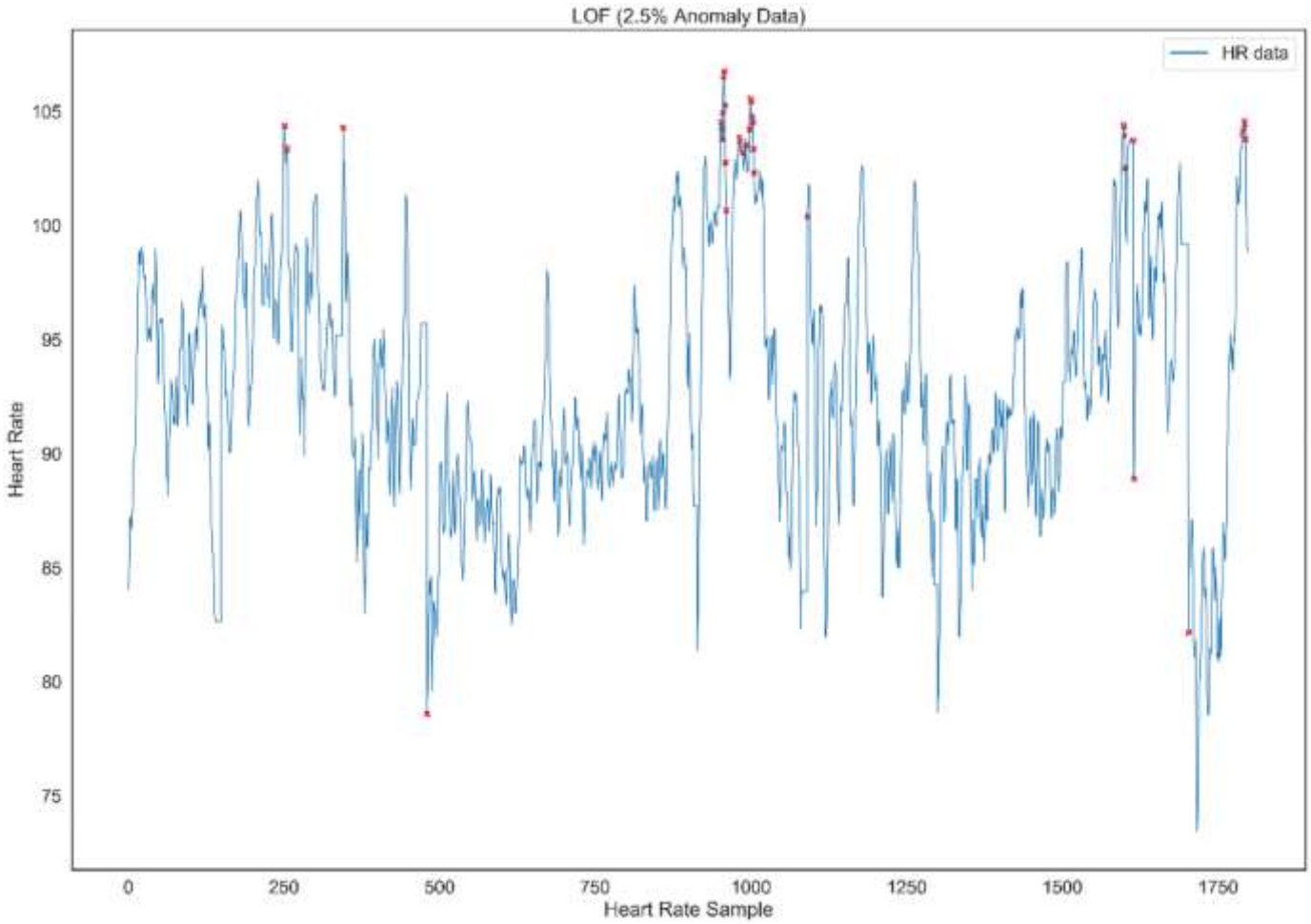
The LOF is equal to the idea that A point is labeled as an outlier if the density around that point is significantly different from the density around its neighbors.

LOF is a metric that reflects the degree of abnormality of the observations and is used to define proximity-based models. It measures the local density deviation of a given data point with respect to its neighbours. The idea is to detect the samples that have a substantially lower density than their neighbours. This density is calculated by ratio involving the k-nearest neighbours algorithm.

Illustration of the LOF principle



Example of Local outlier factor



Isolation forest

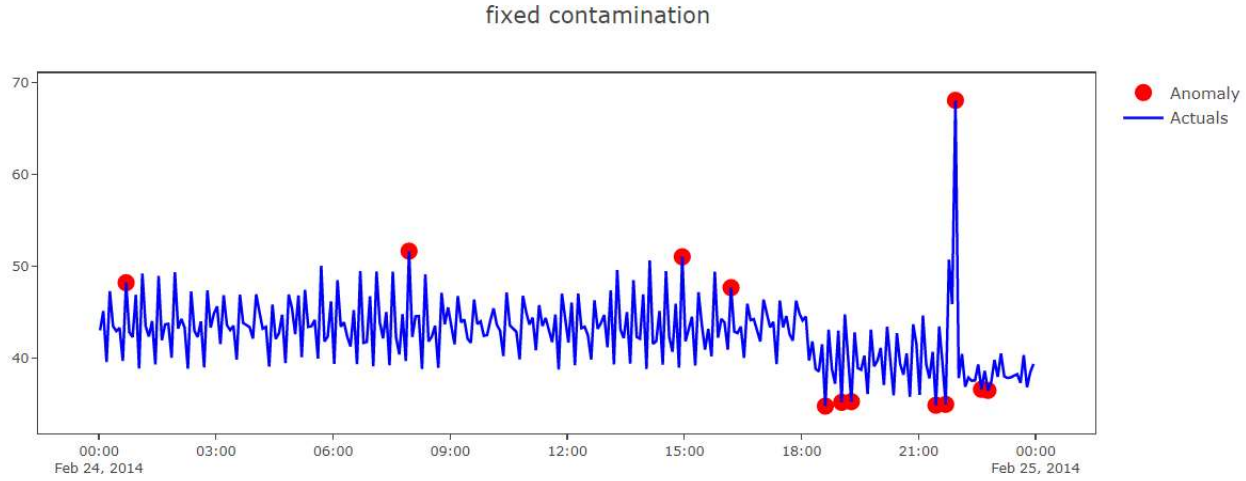
Isolation Forest is a tree ensemble unsupervised method.

- In difference with supervised decision tree-based methods here each partition perform randomly.
- In Each tree for randomly selected segment or features the partition is carried out also randomly in the range between minimum and maximum values of selected feature or segment.
- The anomaly selection is based on the assumption that a normal point requires more partitions to be identified in average (by all trees) than an abnormal point.
- As with other outlier detection methods, an anomaly score is required for decision making.

Isolation Forest, like any tree ensemble method, is built on the basis of decision trees. In these trees, partitions are created by first randomly selecting a feature and then selecting a random split value between the minimum and maximum value of the selected feature. Thus outliers are less frequent than regular observations and are different from them in terms of values (they lie further away from the regular observations in the feature space). That is why by using such random partitioning they should be identified closer to the root of the tree.

Isolation Forest is an unsupervised machine learning algorithm for identifying anomalies within a dataset by isolating anomalies as they are few and different.

Example of Isolation forest work



1.4.3.2 Semi-supervised methods

Z-score

If normal samples are known the **Global threshold** method can be clarified such as

Input data can be transformed to the so-called z-score.

$$z(n) = \frac{y(n) - EV(y_{norm})}{std(y_{norm})},$$

where $EV(y_{norm})$ and $std(y_{norm})$ are mean value and standard deviation for normal data.

- All the z values above 2.5 or below -2.5 should be considered as outliers with confidence 99%.

The z-score global threshold estimation can be done for sliding windowed data.

If the data are not normal or have small size the modified (more robust) z-score can be taken as

$$z_{modif} = 0.6745 \frac{y(n) - median(y_{norm})}{MAD(y_{norm})},$$

where $median(y_{norm})$ and $MAD(y_{norm})$ are median value and mean absolute deviation for normal data.

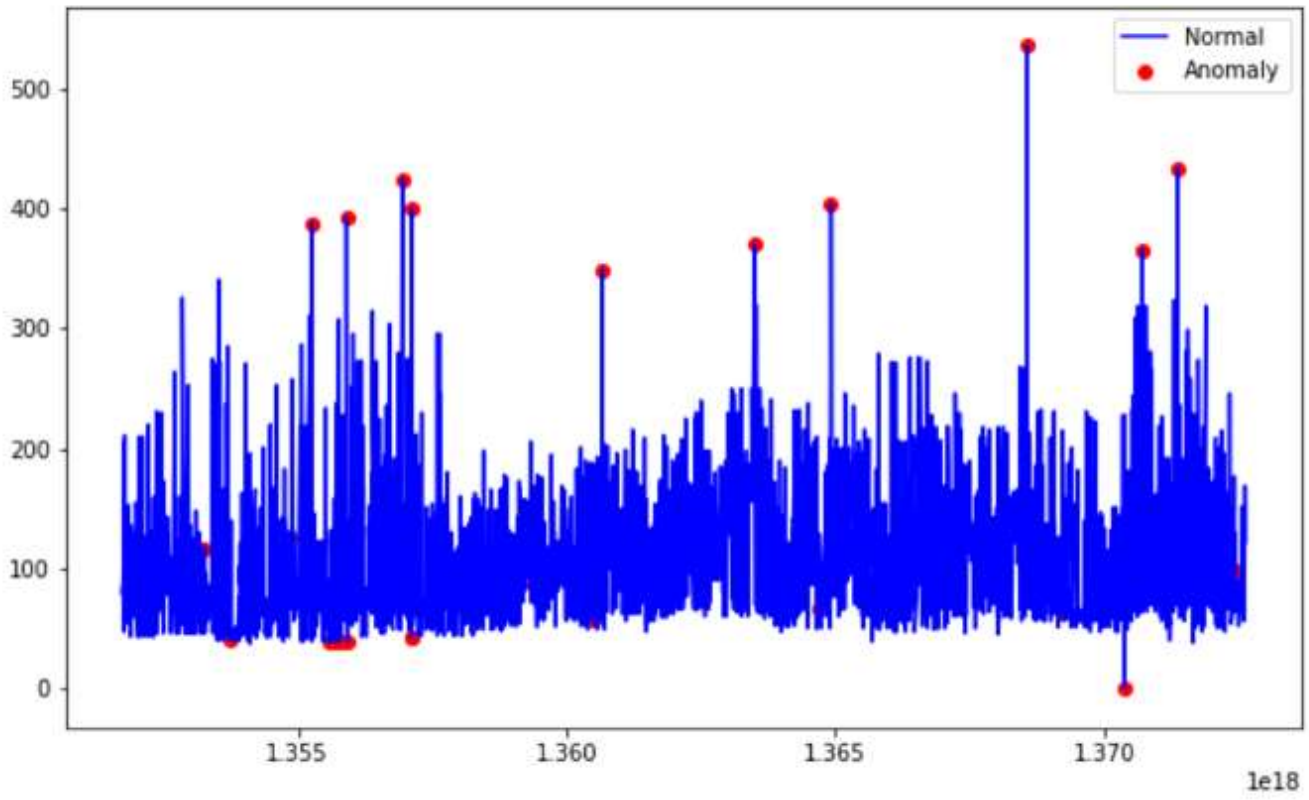
One-Class Support Vector Machines, OSVM

- The method is based on the idea of build only one support vector (or one non-linear decision boundary) that divide normal data (labeled as normal) from another.
- The OSVM require to use kernel-trick, in general with RBF kernel. It is necessary to build a spherical boundary around the labeled data.
- The solution of the OSVM problem is the hyperplane (support vector) that maximizes the distance from labeled data to the origin point (zero point).

In this case the volume of this hypersphere is minimized, to minimize the effect of incorporating outliers in the solution.

- The method is very sensitive for dirty in the labeled data.

> Actually method can be considered as one class vs rest task. In OSVM we are implicitly assume that after kernel trick abnormal data will be highly spreaded in the near-zero - point area.

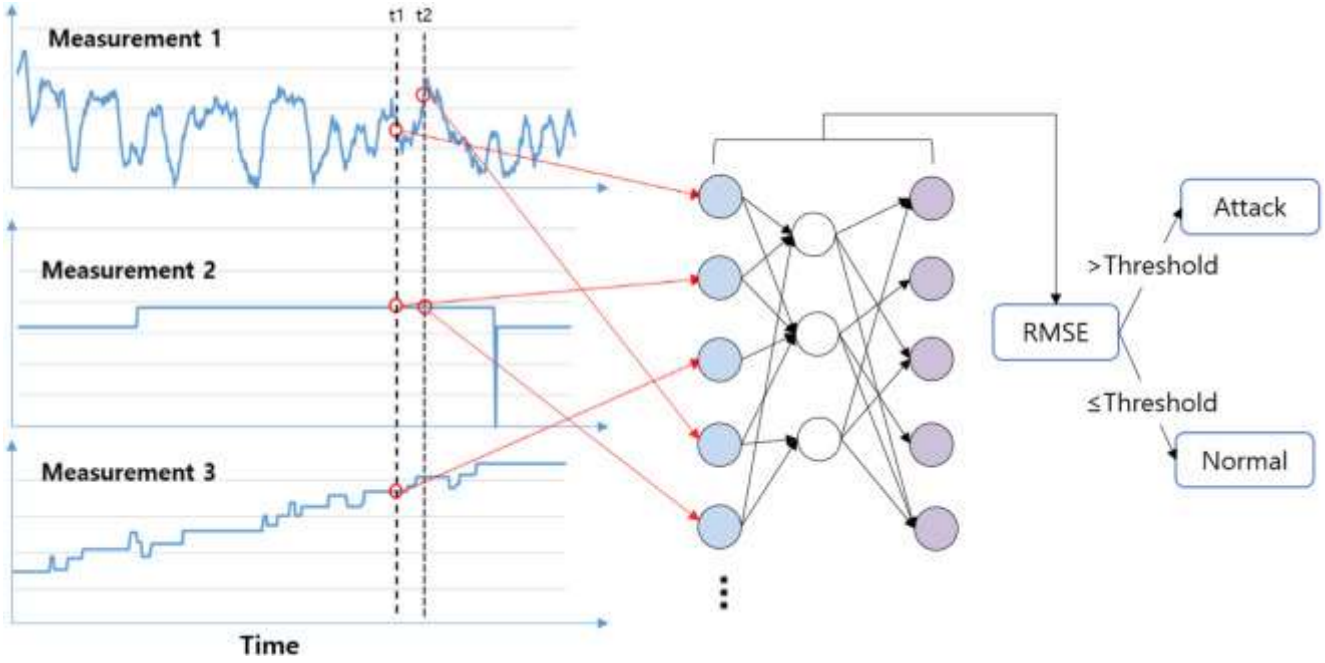


Autoencoder based anomaly detection

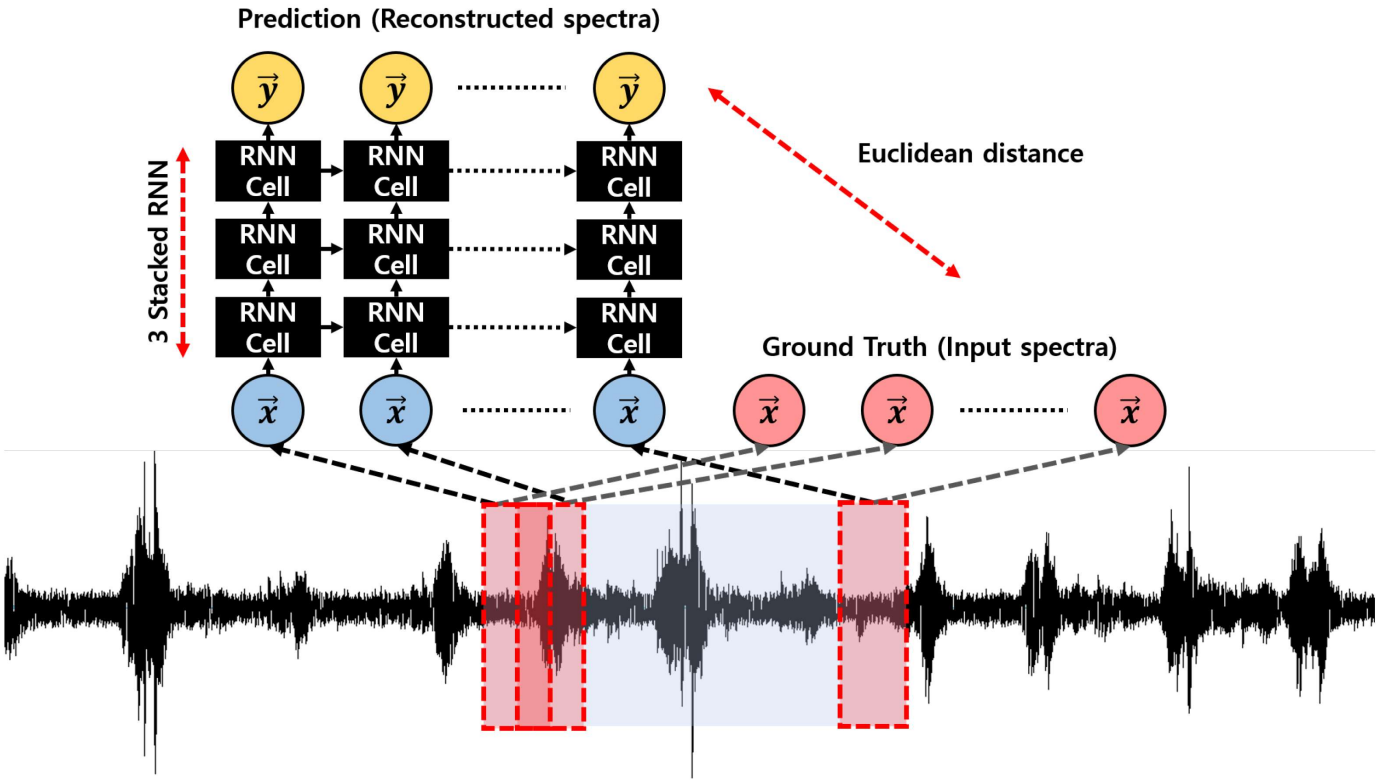
- The error of the trained autoencoder reconstruction result similar to the Anomaly Score.
- The network is training on the data the most of them are assume to be normal.
- The method can be considered in the unsupervised and semi-supervised tasks.

  AE is mainly used for feature extraction and dimension reduction as unsupervised method. But it also good for anomaly detection problems.

  Autoencoder consists of encoding and decoding parts. In encoding part, main features are extracted which represents the patterns in the data, and then each samples is reconstructed in the decoding part. The reconstruction error will be minumum for normal samples. On the other hand, the model is not able to reconstruct a sample that behaves abnormal, resulting a high reconstruction error. So, basically, the higher reconstruction error a sample has, the more likely it is to be an anomaly.



Example of RNN-based autoencoder based anomaly detection.



Note In the anomaly detection task Autoencoder can be considered as semi-supervised task if training is performing only on normal data.

1.4.3.3 Supervised methods

Supervised Neural Network and other machine learning

The supervised task can be solved in two ways

1. Based on the deviation all data into two classes - normal data and abnormal data.
2. Based on the regression task - by estimation the error of prediction, using supervised samples.

Note

In the regression-like case the anomaly detection can be also considered as semi-supervised task.

The popular methods of the supervised machine-learning based anomaly detection are:

- *SVM*
- *Random Forest*
- *k-Nearest Neighborhoods*
- *Neural-Network.*

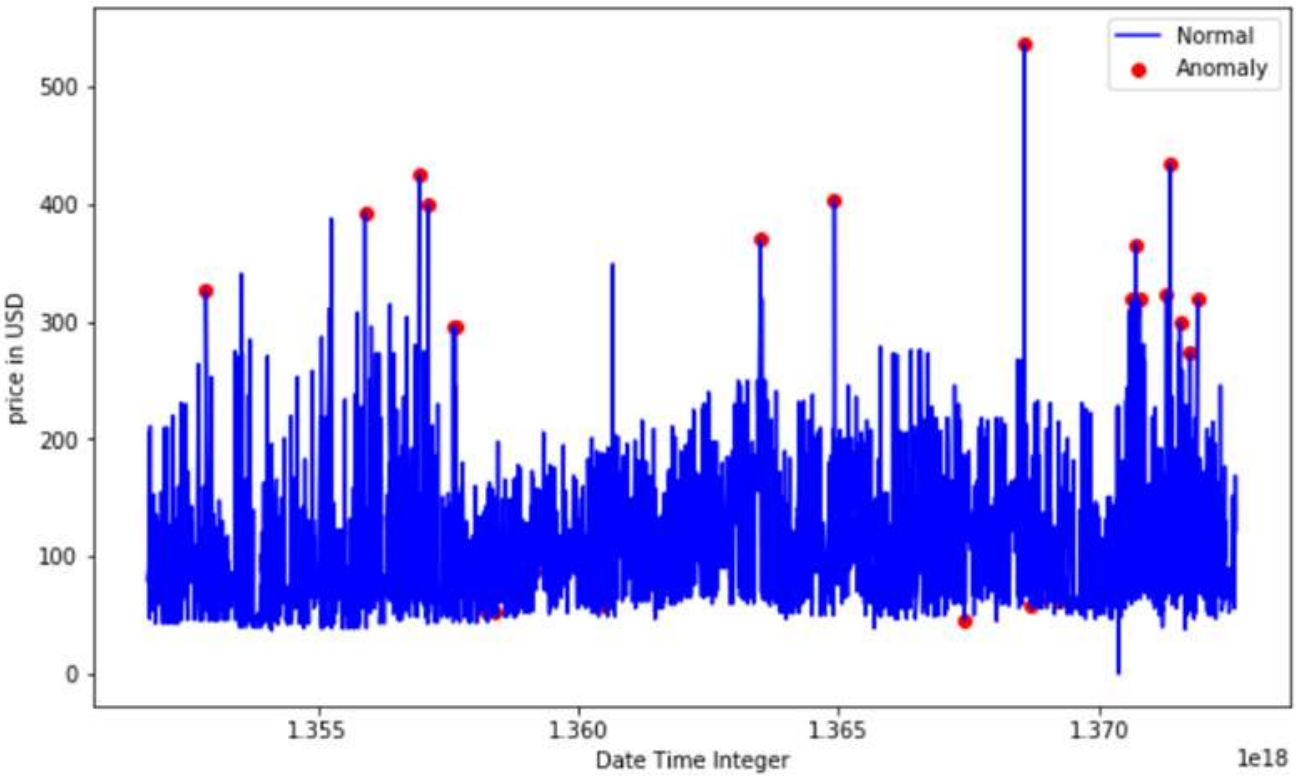
Gaussian Distribution (Eliptic-based method)

- The method is based on the assumption that data are normally distributed.
- Thus outliers are lay on the tails of the probability distribution.
- The tail part can be set by choose quantiles.
- In the case of supervised learning distribution can be calculated by distances between point.
- As a rule Mahalanobis distance is using.

Gaussian Distribution method can be considered as unsupervised method - same as the Global threshold method. Elliptic Envelope is intuitively built on the premise that data comes from a known distribution. If we draw an ellipse around the gaussian distribution of data, anything that lies outside the ellipse will be considered an outlier.

Analogue of Knn with Mahalanobis distance

Example of Eliptic-based method work for Time series.



Example of Mlahanobious distance distribution for data

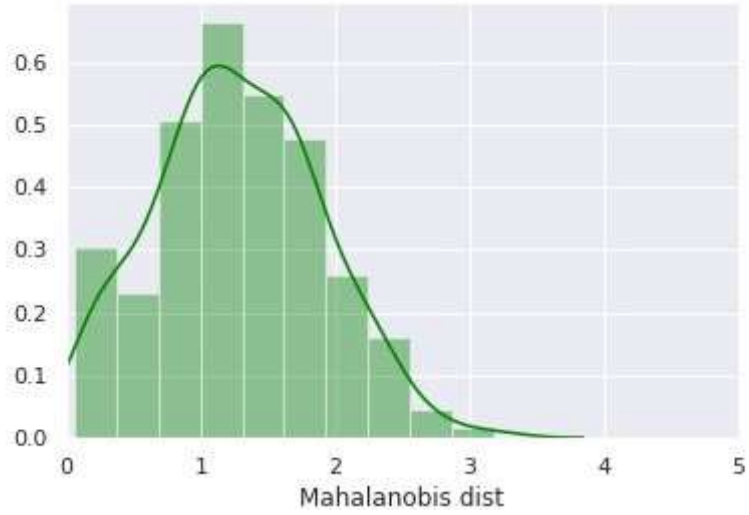


Illustration of the LOF, OSVM, IF and Eliptic methods

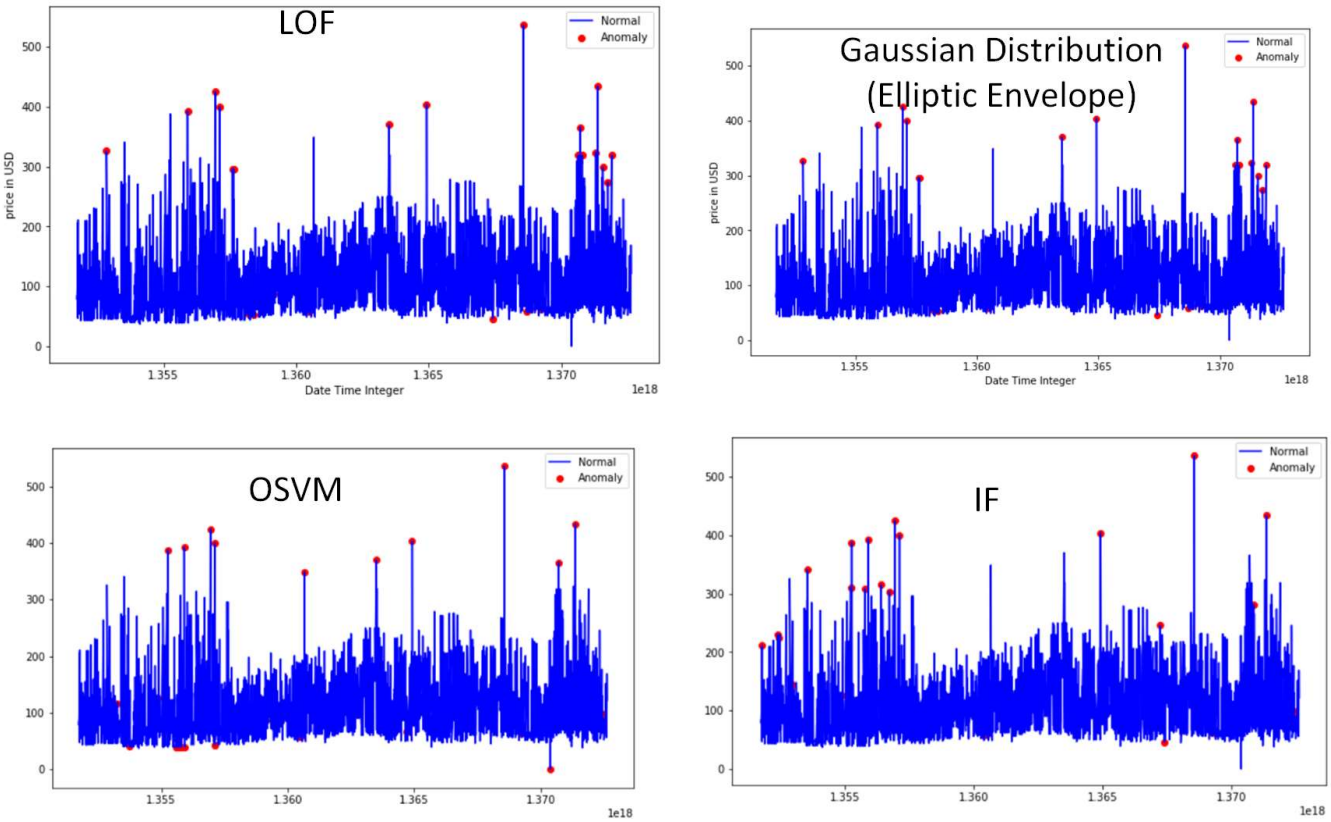
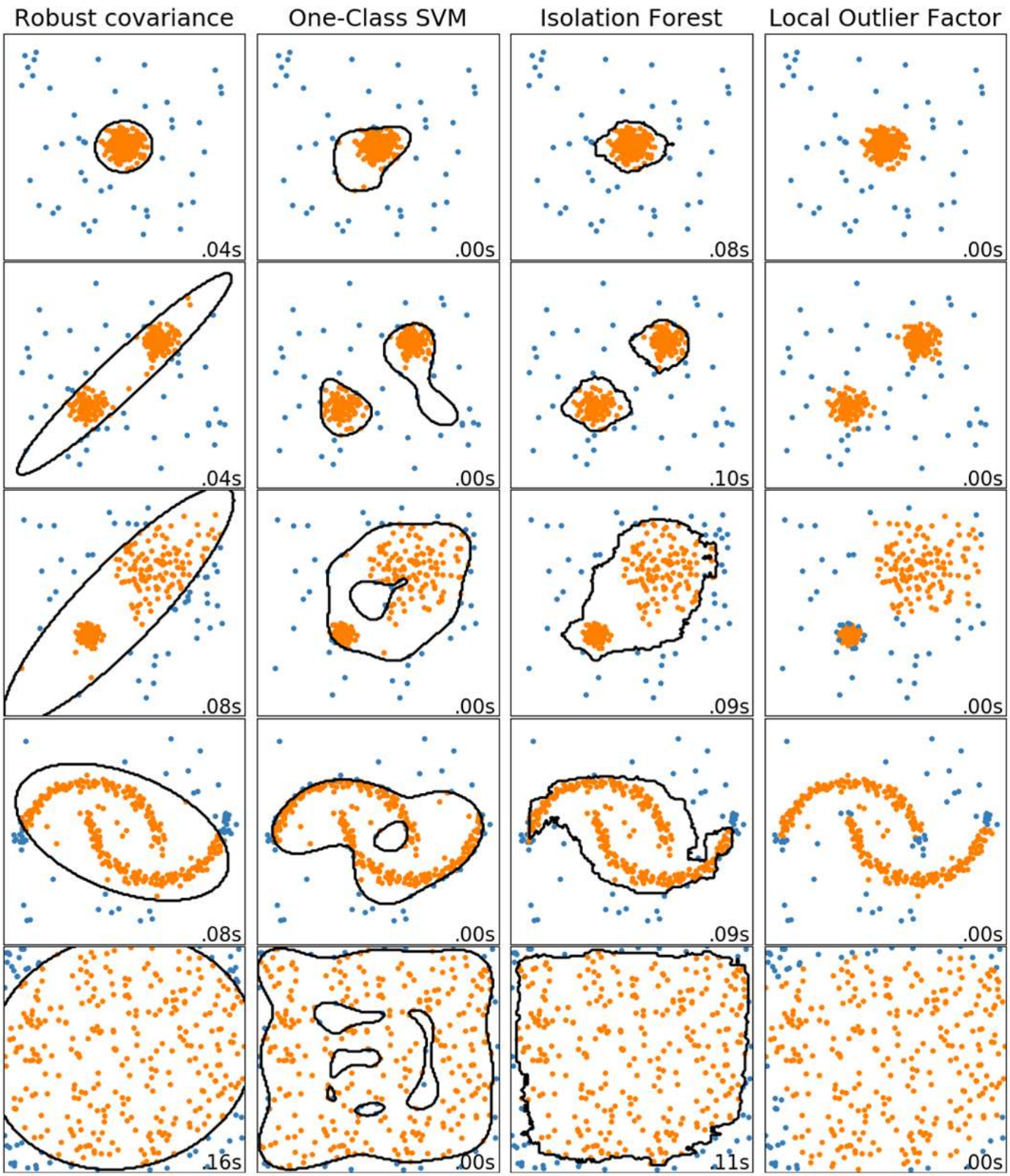


Illustration of OSVM, IF and LOF principles of work

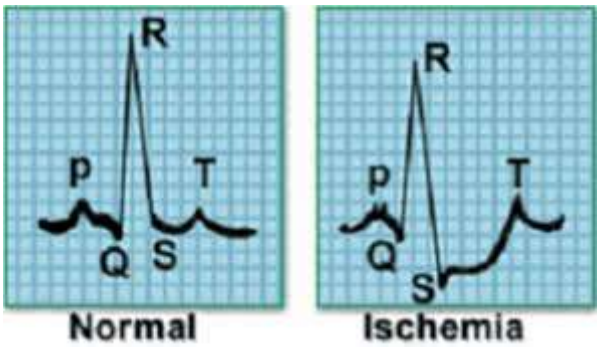


1.5 Supervised Classical ML

1.5.1 Machine-learning methods for TS classification

There are two main tasks of using supervised machine learning for time series analysis.

- Regression (prediction, parameter estimation);
- Classification.



A common, but problematic solution to time series classification is to treat each time point as a separate feature and directly apply a standard learning algorithm (e.g. scikit-learn classifiers).

In this approach, the algorithm ignores information contained in the time order of the data.

One of the most popular and traditional Time Series Classification approaches is the use of a **k-nearest neighbor classifier coupled with a distance function**.

Particularly, the Dynamic Time Warping (DTW) distance.

This classifier has been shown to be a very strong baseline (**kNN-DTW method**).

> The more accurate is to use ensembling of the individual classifiers with different distance measures. It is shown that this approach in general outperforms all of the ensemble's individual components.

Beside kNN-DTW there are a plenty of Time Series Classification Methods.

The main drawbacks of kNN-DTW are the following.

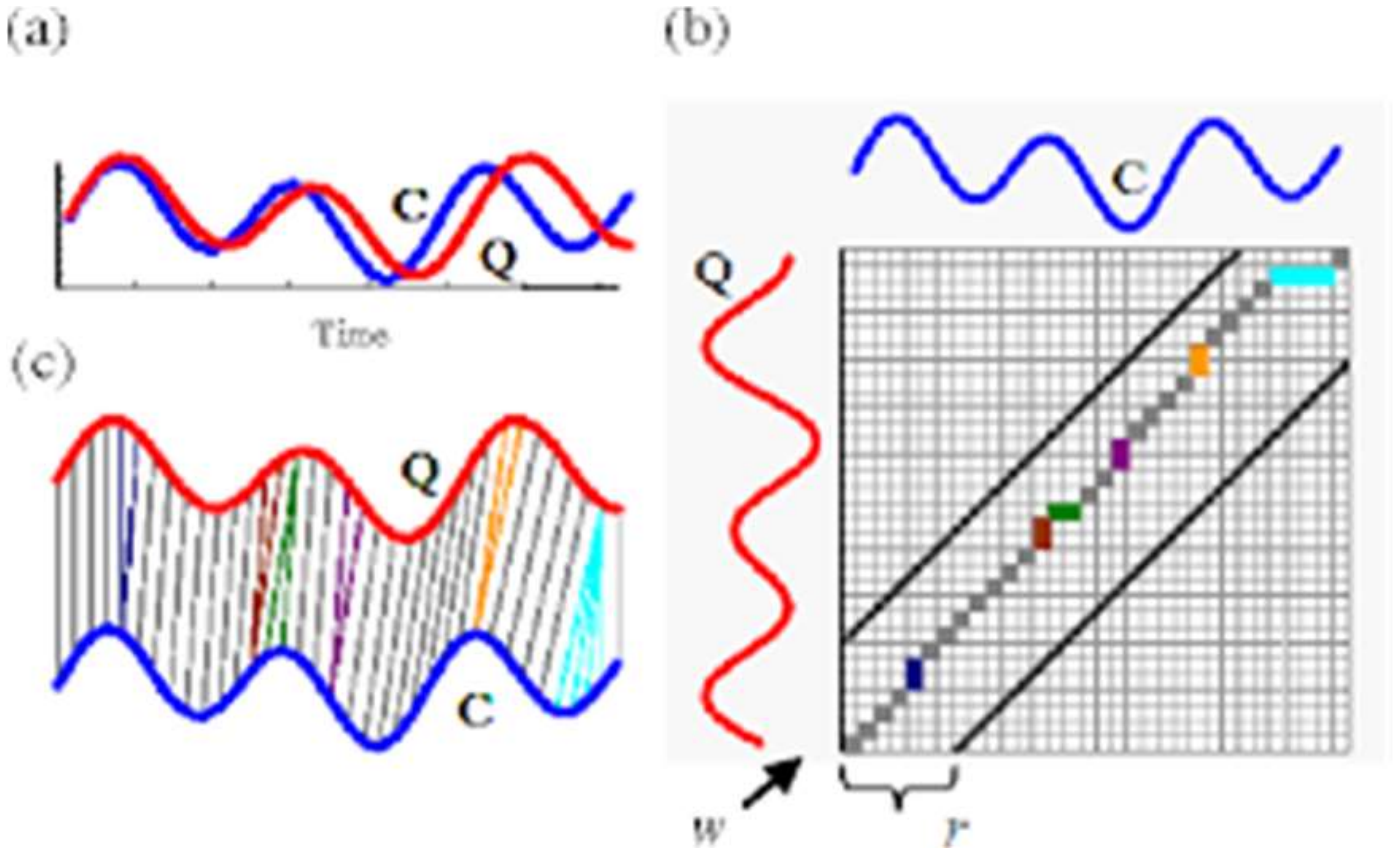
- the method requires a lot of space and time to compute.
- During classification, the kNN-DTW compares each object with all the other objects in the training set.

- KNN provides limited information about why a series was assigned to a certain class.
- The noise in a series may overpower subtle differences in shape that are useful for class discrimination.
- The kNN-DTW algorithm has achieved an outstanding performance on the benchmark datasets, however it suffers from $O(n^2 \cdot T^4)$ time complexity.

Most of the time series classifiers are:

- **Distance-Based Classification.**
- **Ensemble of decision trees (random forest, Time Series Forest).**
- **Support Vector Machine.**
- **XGBoost.**
- **Bag of Symbols Dictionary (BOSS).**

Example of kNN-DTW Classifier



A time series forest (TSF)

TSF classifier adapts the random forest classifier to series data.

TSF is a Interval-based classifier.

- Split the series into random intervals, with random start positions and random lengths.
- Extract summary features (mean, standard deviation, slope and e.t.c.)

  from each interval into a single feature vector.
- Train a decision tree on the extracted features.
- Repeat steps 1–3 until the required number of trees have been built or time runs out.
- New series are classified according to a majority vote of all the trees in the forest.

  In a majority vote, the prediction is the class that is predicted by the most trees of the forest.

- Experimental studies have demonstrated that time series forest can outperform baseline competitors,
    such as nearest neighbors with dynamic time warping.
- Time series forest is also computationally efficient.
- Last, time series forest is an interpretable model.
    Time feature importance can be extracted from time series forest.

Random Interval Spectral Ensemble (RISE)

RISE is a popular variant of time series forest.

In difference with TSF RISE:

1. uses a single time series interval per tree
2. it is trained using spectral features extracted from the series, instead of summary statistics.

RISE use several series-to-series feature extraction transformers, including:

- Fitted auto-regressive coefficients;
- Estimated autocorrelation coefficients;
- Power spectrum coefficients (the coefficients of the Fourier transform);

The RISE algorithm is straightforward:

1. Select random interval of a series (length is a power of 2).

   (For the first tree, use the whole series)

2. For the same interval on each series, apply the series-to-series feature extraction transformers

   (autoregressive coefficients, autocorrelation coefficients, and power spectrum coefficients).

3. Form a new training set by concatenating the extracted features.

4. Train a decision tree classifier

5. Ensemble 1–4

Class probabilities are calculated as a proportion of base classifier votes.

> RISE controls the run time by creating an adaptive model of the time to build a single tree. This is important for long series (such as audio), where very large intervals can mean very few trees.

Shapelet-Based Classifiers

Shapelets are subsequences, or small sub-shapes of time series that are representative of a class.

Shapelet-based classifiers search for shapelets with greatest discriminatory power.

In the training set the presence of certain shapelets make one class more likely than another.

In the training data each shapelet considered is evaluated according to some information gain criteria (e.g. entropy). The strongest non-overlapping shapelets are retained.

> Shapelets can be used to detect "phase-independent localised similarity between series within the same class".

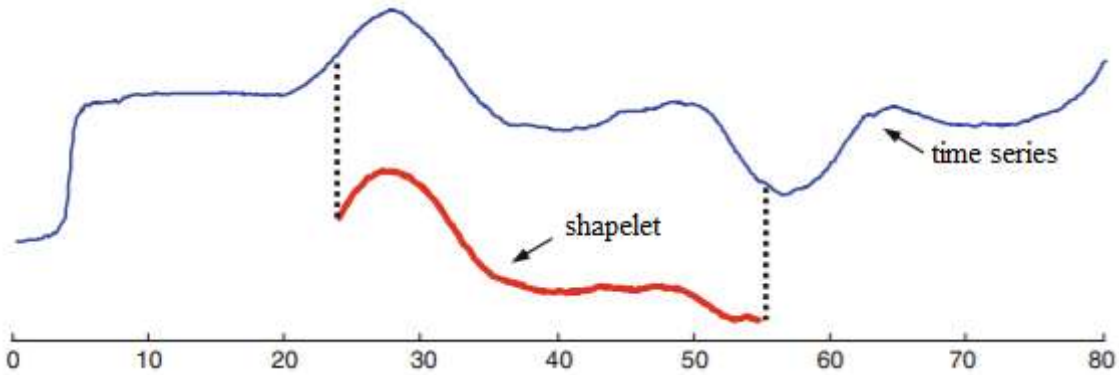
Informally, if we assume a binary classification setting, a shapelet is discriminant if it is present in most series of one class and absent from series of the other class.

To assess the level of presence, one uses shapelet matches:

$$d(\mathbf{y}, \mathbf{s}) = \min_n \|\mathbf{y}_{n \to n+L} - \mathbf{s}\|_2$$

where L is the length (number of timestamps) of shapelet s and $\mathbf{x}_{n \to n+L}$ is the subsequence extracted from time series y that starts at time index n and stops at $n + L$.

If the above-defined distance is small enough, then shapelet s is supposed to be present in time series y.



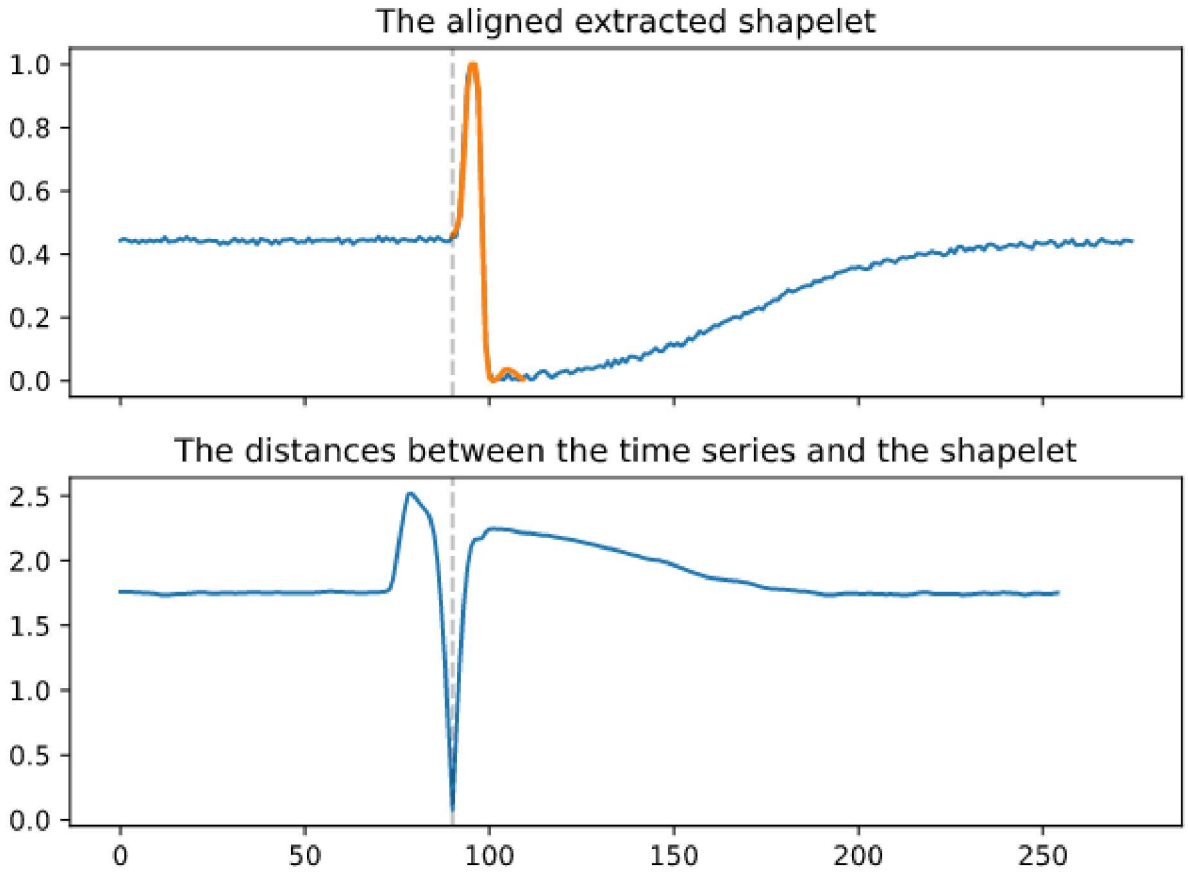
Shapelet Transform Classifier

- The algorithm first identifies the top k shapelets in the dataset.
- Then k features for the new dataset are calculated.

Each feature is computed as the distance of the series to each one of the k shapelets.

- Finally, any classification algorithm can be applied to the shapelet-transformed dataset (e.g. random forest).

  Before classification the feature extraction of shapelets can be performed (for instance by PCA).



Dictionary-Based Classification

The methods are based on the following steps.

- Transform the series into segments or by sliding window with some step.

- For each of segment (or by sliding across segment with window) build a one or bag of words, for instance by quantization, rounding to integer or using some specific algorithms.

    In the case of quantization and rounding to integer the resulted "turnicated" values can be corresponded to some virtual and finite dictionary of words

    each word with equal length (with finite number of possible symbols and its possible combinations - words).

- Build a histogram of words meeting amount for each window.
- Finally, any classifier can be trained on the word histograms extracted from the series.

The one of the most popular method for dictionary obtaining is

Bag of SFA Symbols (Bag of Symbolic Fourier Approximation (SAF) Symbols, BOSS)

Word features for BOSS classifiers are extracted from series using the

Symbolic Fourier Approximation (SFA) transformation:

- Calculate the Fourier transform of the window (the first term is ignored if normalization occurs).
- Quantization the first *l* Fourier terms into symbols to form a "word".

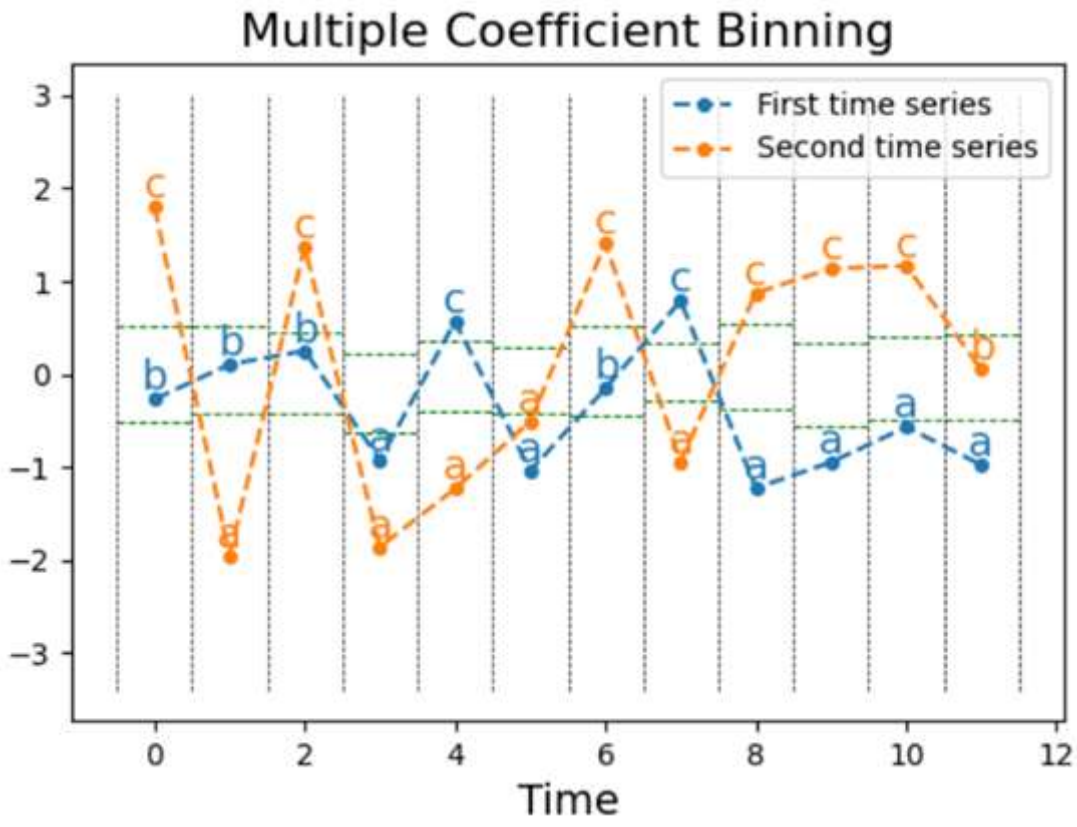
    for instance round to the integer part or using **Multiple Coefficient Binning (MCB).**

        MCB is a supervised algorithm that bins continuous time series into a sequence of letters.
- A dictionary of these words is constructed as the window slides, recording a count of each word frequency.

    If the same word is produced by two or more consecutive windows, the word will only be counted

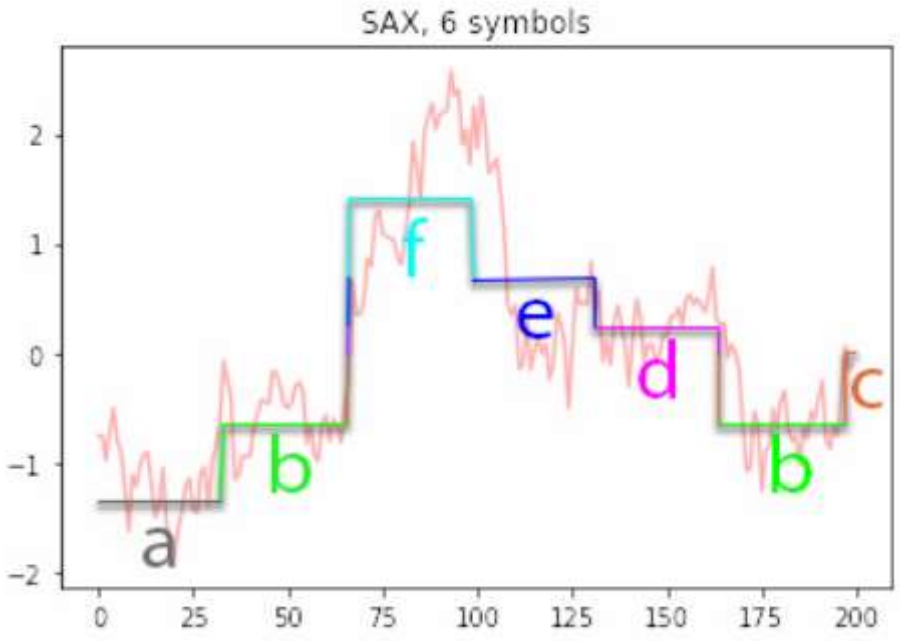
once.

Example of Dictionary transforms for two segments.

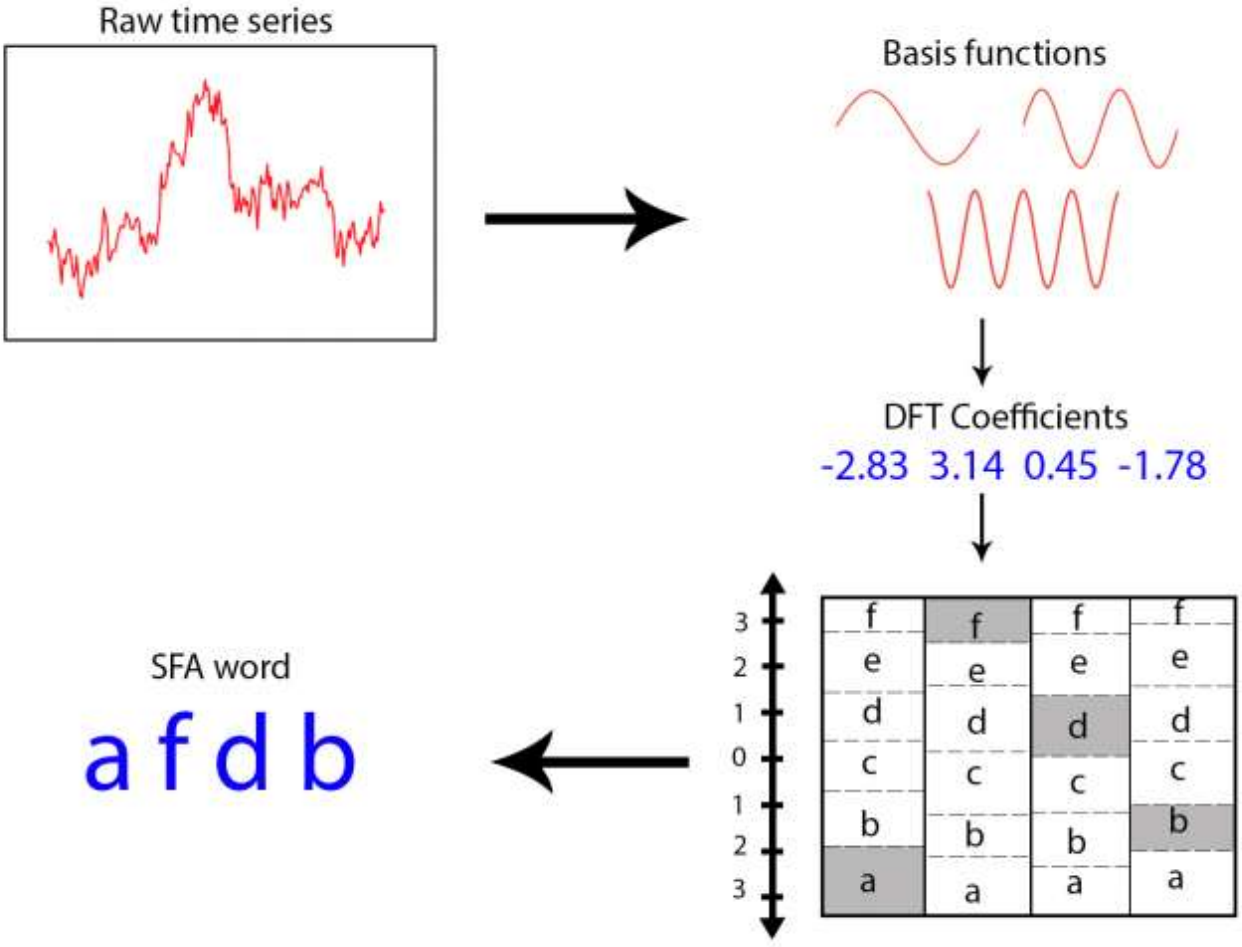


- 1 class word: bbbacabcaaaa
- 2 class word cacaaacacccb

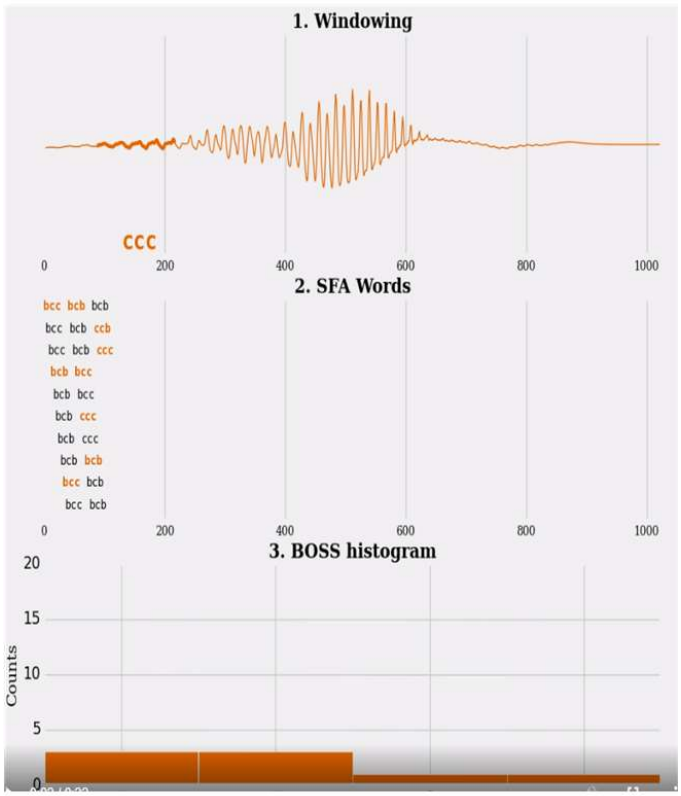
SAX

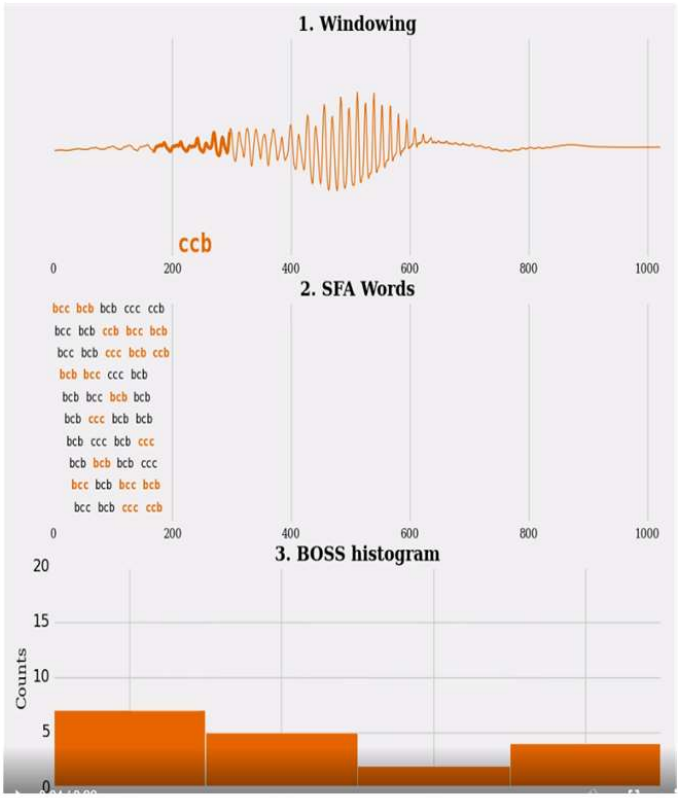


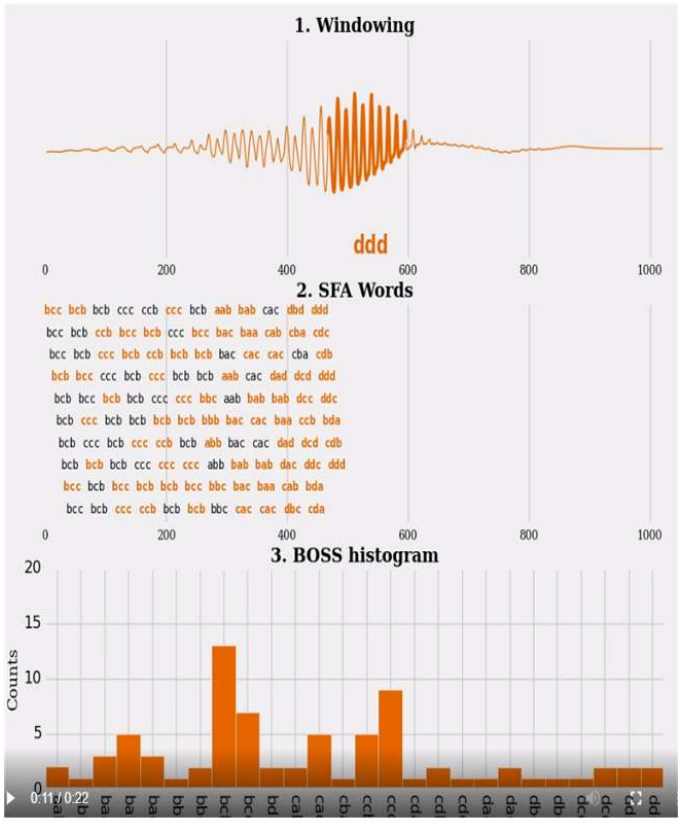
SFA

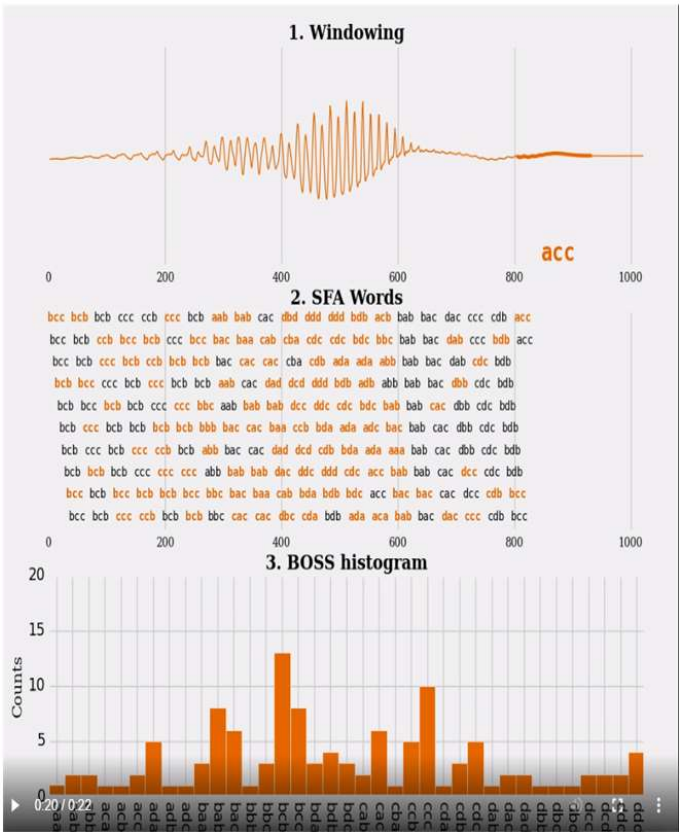


Example of BOSS work for sliding window across the one segment!.









Dynamic Time Warping Distance based knn Classifier

For time series, the most popular k-nearest-neighbours algorithm is based on dynamic time warping (dtw) distance measure.

The **DTW algorithm** has the follows steps:

1. Calculate the distance between the first point in the first series segment and every point in the second series.

   Select the minimum of the calculated values and store it (this is the "time warp" stage).
2. Move to the second point and repeat stage 1.

   Move step by step along points and repeat stage 1 till all points are exhausted.
3. Calculate and Select the minimum of distances between the first point in the second series segment and every point in the first series.
4. Move step by step along points in the second segment and repeat stage 3 till all points are exhausted.
5. Sum all the stored minimum distances.

ROCKET

ROCKET Classifer

ROCKET Classifer is type of the Shapelets-based classifers is based on the so-called **ROCKET transforms**.

- The **ROCKET transforms** are the time series transforms using random convolutional kernels (random length, weights, bias, dilation, and padding).
- ROCKET Classifer computes two features from the resulting feature maps (after transformations): the max value, and the proportion of positive values to all (ppv).
- The transformed features are used to train a linear classifier.

HIVE-COTE

The Hierarchical Vote Collective of Transformation-based Ensembles (HIVE-COTE)

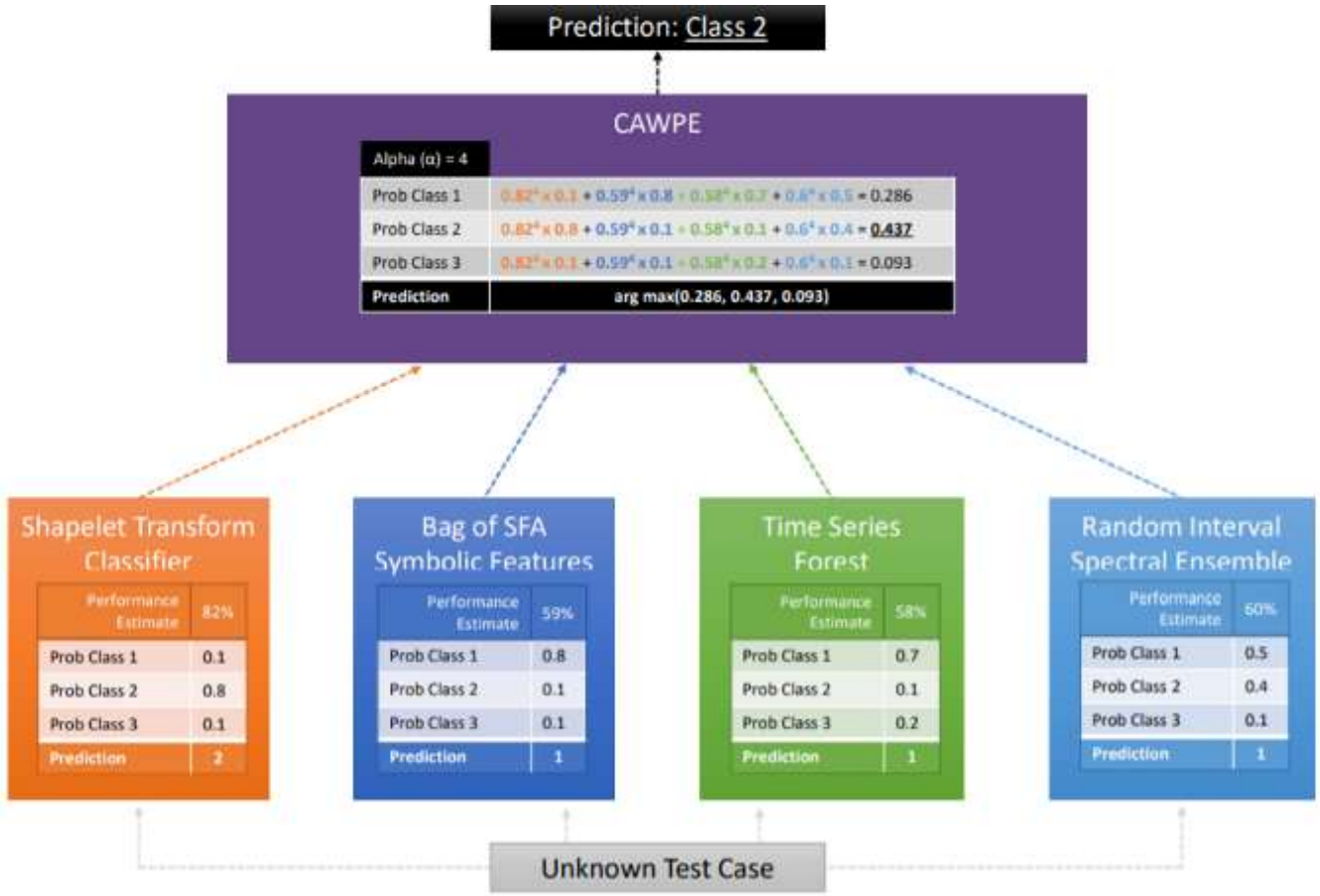
is a meta ensemble built on the classifiers discussed previously.

HIVE-COTE predictions are a weighted average of predictions produced by its members:

shapelet transform classifier, BOSS, Time Series Forest, and RISE.

Each sub-classifier estimates the probability of each class. The control unit then combines these probabilities (CAPWE).

The weights are assigned as the relative estimated quality of the classifier found on the training data.



1.5.2 Time series forecast

There are several types of the time series forecast problem solution

Stochastic models (or model based approach).

The methods can be divided on non-parametric and parametric.

- **non-parametric** such as moving average, Holt-Winter, simple regression:
  - based on the analytical model of the series behavior,
  - easy to train and retraining,
  - low probability of overfitting,
  - provide the best accuracy for comparably simple data (stationary with Gaussian noises or some simple noises, like symmetric-distributed ones).
  - accuracy highly depends on the amount of data for training.
  - the performance dramatically decrease if the data behavior are differs from the assumed one (form the assumed statistical hypothesis).
  - Work well only for univariate case.
  - Can be easy interpreted.
  - Can have analytical solution (for linear regression fore instance).
  - For it work wee need to know distribution type or make strong assumption about it.
- **parametric** such as ARIMA, GARCH, Prophet and e.t.c.
  - based on the analytical-parametric model of the series behavior,

- simple for understanding and implementation,
- easy to train and retraining,
- provide sufficient high accuracy of prediction,
- better to work with univariate data,
- better with relatively simple non-stationarity data,
- accuracy highly depends on the amount of data for training,
- accuracy highly depends on the choose of the hyperparameters values (here hyperparameters are analogue of the statistical hypothesis in the non-parametric approach),
- cannot handle the complex non-linear relationship among the data,
- worth performance for huge multivariate non-stationarity series (especially in the case of non-normalized data with different in behavior and e.t.c).
- Work well only for univariate case.
- Can be easy interpreted.
- More robust to inferences in the case of well selected model (due to super resolution with inferences for small datasets).
- low performance on the large datasets with complex relation (missing values, outliers, nonuniform discretization).

Data-driven models (or machine learning approach)

The methods can be divided on classical machine learning and deep neural network.

- **Classical data-driven models (or machine learning approach)**.

  Such methods as Support Vector Regression (SVR), Regression Random forest, XGBoost, and e.t.c.

  - Allow to work with highly non-linear data.
  - No need of statistical hypothesis for model.
  - Good at handling non-stationary relationships among data.
  - Easy to train.
  - Accuracy highly depends on the choose of the hyperparameters values.
  - Implicitly dependence of the chose model and data on the prediction results.
  - Accuracy highly depends on the similarity between trained data and inference one.
  - Hard to achieve comparable accuracy with model based approach for relatively simple data.

- **Deep Neural Network**.

- No need of statistical hypothesis or a specific model form.
- Can approximate any function with skipped data, anomalies and other irregular patterns.
- Allow to work with huge multivariate data series with complex relation of behavior among data.
- Automatically extract and handle complex features and dependency relationships.
- Require hard tuning of the hyperparameters.
- Frequently the ensemble of the networks need to achieve high accuracy.
- Hard to retraining.
- Hard to achieve comparable accuracy with model based approach for relatively simple series.

The choose of the specific methods is depends on the task.

For simple and univariate data the model based approach are recommended.

For complex and multivariate data in huge amount the Data-driven models could provide better performance.

Stochastic model

Prophet

The facebook business forecast adaptive model

Standard methods like ARIMA or Holt-Winter generate stable forecasts, however they have difficulties forecasting quick changes in time-series, especially when these changes are due to multiple seasonalities or moving rare events (like holydays).

From the other side, if some generalized (parametric) model of process is known the forecast model can be introduce as adaptive parametric one.

For the business forecast like sales, price or demand prediction the model can be described as deterministic trend one with rare change; multiple seasonalities (days, weeks, month, years); and rare events like holydays.

The named above factors can be introduced into the model, which will be these given as:

$$y(x) = trend(x) + seasonal(x) + holydays(x) + noise(x),$$

where

- $trend(x)$ is the non-periodic change
    - piecewise linear trend

$$trend(x) = (k + a(x)\delta)x + (m + a(x)\gamma),$$

where k is the growth rate; δ and γ has the rate adjustments and m is the offset parameter.

- logistic (decrease and increase) with saturation

$$trend(x) = \frac{C}{1 + exp(-k(x - b))},$$

where C, k, b are trainable parameters:

C is the carrying capacity;

k is the growth rate;

b is an offset parameter.

- library includes additional routine to choose point of trend change by history.
- point of trend change can also be set manually.

- $seasonal(x)$ is the regular periodic change
  - 6-x days seasonalities (monday, tuesday, wednesday, thursday, friday, saturday) are modeled as 0 and 1, sunday modeled as linear combination of previous siz days.
  - week, month and years multiple seasonalities modeled as Fourier series

$$seasonal(x) = \sum_{n=1}^{N}(a_n \cos(2\pi nx/P) + b_n \sin(2\pi nx/P)),$$

where P is the period in days (365.25 for year, 7 for week); a_n and b_n parameters need to be estimated.

- The model could be overfitting or underfitting depends on the chose the number of the trend component.

- $holydays(x)$ are rare events, modeled as

$$holydays(x) = Z(x)k,$$
$$Z(x) = [1(t \in D_1), 1(t \in D_2), \dots)],$$

where D_i is the holydays duration; k is the normal distribution with std as parameter and zero value.

- $noise(x)$ are the i.i.d. - Gaussian noises.

Note

In some source you may see the following prophet equation notation

$y(x) = trend(x) + seasonal(x) + holydays(x) + user\_regressor(x) + noise(x$

where $user\_regressor(x)$ are regressors that are does not described by other model (in the equation in most sources, the component is including implicit). However in the previous notation $user\_regressor(x)$ are introduced into $seasonal(x)$ component.

Advantages of prophet

- Using time as a regressor, Prophet is trying to fit several linear and non linear functions of time as components.
- Ability to easily model any number of seasonalities.
- Ability to work with missing dates in time-series.
- Easily integrates holidays in the model.
- Built-in uncertainty modeling with full Bayesian sampling allows for model transparency.
- Allowing flexible step-wise linear or logistic trend modeling with user-specified change points.

Drawbacks of prophet

- Prophet does not allow non-Gaussian noise distribution
- Prophet does not take autocorrelation on residual into account (thus require only Gaussian noise distribution).
- Prophet does not assume stochastic trend behavior (random walk).

    Long-horizon forecasting can be volatile with automatic changepoint selection

In [ ]:

```

```