

Feature generation for multiscale time series forecasting multimodels

Abstract—The paper presents a framework for the multiscale time series forecast. We reduce the problem of time series forecast to the regression problem and propose a method of constructing efficient feature description for the regression problem. The method involves feature generation, followed with dimensionality reduction procedure. Generated features include historical information about the target time series as well as other available time series, local transformations and multiscale features. We apply several forecasting algorithms to the resulting regression problem and investigate the quality of the forecasts for various horizon values.

I. INTRODUCTION

The paper investigates behavior of a device within the concept of Internet of Things. The device at question is monitored by a set of sensors, which produces large amount of multi-scale time series during its lifespan. These time series have various time scales since distinct sensors produce observations with various frequencies (milliseconds, days, weeks, etc). The main goal is to forecast the observations of a device in a given time range.

We assume that the sampling rate of each time series is fixed and each time series has its own forecast horizon. The problem of multi-scale analysis arises in such applications as weather prediction, medical diagnosis and monitoring various sensor time series [1], [2], [3], [4] [medical, medical, traffic, ecology]. Motivation for multi-scale analysis comes from the assumption that the behaviour of complex signals may be governed by essentially different processes at various time scales. Thus, the time series should be modeled separately at each scale. This approach is used in time series classification, prediction and fault detection [5], [3], [6]. Regardless of the goal of multi-scale analysis, it includes sequential averaging of the time series to obtain more coarse-scaled time series [7], or, more rarely, differencing the time series for a more detailed, fine-scaled version of the time series [8]. Averaging and dif-

ferencing, which is equivalent to application of Haar's wavelet transform [8], may be replaced by any other pair of low and high pass wavelet filters [9] or convolution operation with some kernel function [10]. Next steps depend on the goal of multi-scale analysis. Using multi-scale approach in time series prediction usually involves determining optimal scales [10], [2], decomposition of time series into separately forecasted components and combination of the obtained forecasts.

II. RELATED WORK

Along with generic methods of time series forecasting, such as Autoregressive Moving Average Models (ARMA), Autoregressive Integrated Moving Average Models (ARIMA), many authors report high predictive performance of the methods, originally developed for classification or regression, applied to forecast time series. The latter include Support Vector Regression [11], [12], [13], random forests [14], [15] and artificial neural networks [16], [17]. Random Forests combine decision trees with randomly generated nodes to increase the accuracy of classification or regression [18]. In case of regression trees, each node of the tree splits the input space into two subspaces and each leaf specifies a distinct regression model, which is used for prediction if the input is found in the corresponding region of the input space. Predictions of the trees in the forest are averaged, or, for the probabilistic random forest, the probabilities of the outputs are averaged. The advantage of random forests is their efficiency in case of highly dimensional data due to the randomness incorporated into selecting informative features. Since random forests are essentially ensembles of weak learners, they enjoy high generalization ability, associated with boosting algorithms. The authors of [18] compare regression with probabilistic Random Forest with

Here the input variables are the delayed observations of the time series, and the output is the forecasted value of time

Table I: Notation.

$\mathcal{D} = \{\mathbf{s}^{(m)} m = 1, \dots, M\}$	large set of time series
$s(t)$	the underlying signal
$\bar{\mathbf{s}}$	special case of time series \mathbf{s} for classification
t_{\max}	the “current” time
G, G_s	initial and resampled grid
T_s, T_{rs}	initial and resampled number of observations in \mathbf{s}
$f_s = \frac{1}{\tau_s}$	sampling rate
Δt_p	local prehistory
Δt_r	defines time period (requested) for prediction
$[\mathbf{y}_i^{(m)} \mathbf{x}_i^{(m)}]$	local (t_i) segment from the time series $\mathbf{s}^{(m)}$
$\mathbf{X}^*, \mathbf{x}_i^* = [\mathbf{y}_i \mathbf{x}_i]$	design matrix, $\mathbf{y}_i = [\mathbf{y}_i^{(1)}, \dots, \mathbf{y}_i^{(M)}]$, $\mathbf{x}_i = [\mathbf{x}_i^{(1)}, \dots, \mathbf{x}_i^{(M)}]$
\mathbf{Y}, \mathbf{X}	target matrix, feature matrix
q	number of objects \mathbf{x}_i^* extracted from \mathcal{D}
r	dimensionality of the target vector $[\mathbf{y}_i^{(1)}, \dots, \mathbf{y}_i^{(M)}]$
n	dimensionality of the feature vector $[\mathbf{x}_i^{(1)}, \dots, \mathbf{x}_i^{(M)}]$
$\mathbf{x}^* = [\mathbf{y} \mathbf{x}]$	a testing object?
$\mathcal{J} = 1, \dots, n$	feature index set
$\mathcal{A} \subset \mathcal{J}$	the subset of feature indices
$\mathbf{f}_k(\mathbf{w}_{\mathcal{A}_k}, \mathbf{x}), k = 1, \dots, K$	regression model, defined for a subset \mathcal{A}_k
$\mathbf{w}_{\mathcal{A}_k}$	model parameters (alternatively, \mathbf{w}_k)
$\boldsymbol{\pi} = [\pi_{i1}, \dots, \pi_{iK}]$	vector of correspondence
$\mathcal{I} = \{1, \dots, m\} = \mathcal{B}_0 \sqcup_k \mathcal{B}_k$	set of object indices
\mathcal{B}_0	test set
$\mathcal{B}_k, k = 1, \dots, K$	training sets
Q	quality function
S	error function
$g = g(\mathbf{b}, s)$	parametric functions for feature transformation
$\mathcal{G} = \{g : \mathbf{s} \rightarrow \phi\}$	set of feature transformations
$\phi = [\phi^1, \dots, \phi^M]$	generated feature vector
$\boldsymbol{\varepsilon} = \mathbf{y} - \hat{\mathbf{y}}$	vector of residuals
$\mathbf{c}_1^{(m)}, \dots, \mathbf{c}_P^{(m)}$	k -means centroids
\mathbf{H}^*	Hankel matrix
$\boldsymbol{\theta} = [\mathbf{w}_1, \dots, \mathbf{w}_k, \boldsymbol{\pi}, \beta]$	parameters of the mixture model
z_{ik}, γ_{ik}	latent (correspondence) variables and their expectations (mixture models)

series. However, the authors of [12] show that this prediction framework suffers from systematic error that does not converge to zero as the sample size increases, since both the inputs and the outputs are noised and regression algorithms do not handle the noise in the input correctly. To ensure error convergence, the authors first apply cubic spline approximation, which yields much lower RMSE in case of noisy data. The impact of spline approximation is not so great when the noise is low or discretization is too coarse.

To extend this one-step-ahead forecasting scheme to the case of multiple predictions, one may use iterative, direct or multiple output strategies [19]. Within the iterative strategy, one-step-ahead forecasts are computed recursively, with the newly predicted values of the time series used as the actual future records. A less prone to error accumulation, though

more time consuming method is the direct strategy, which involves estimation of h models to predict h future values of the time series [20]. Finally, the multiple input multiple output (MIMO) strategy allows to obtain h prediction with at one step. In case of SVR, MIMO strategy is based on multivariate SVR [21]. The paper [19] compares different strategies of multi-step-ahead prediction in SVR-based forecasting: direct, iterative and multiple output. Regardless of the horizon values, direct and MIMO strategies consistently achieve more accurate forecasts, than the iterative strategy, with MIMO being most accurate in most cases.

Additionally, there are multiple suggestions on how to combine these forecasting methods [22], [23] or use them in the multi-scale fashion [9], [24], [5], [25], [26].

III. PROBLEM STATEMENT

Consider a large set of time series $\mathcal{D} = \{s^{(m)} \mid m = 1 \dots, M\}$, where each real-valued time series s

$$s = [s_1, \dots, s_i, \dots, s_T], \quad s_i = s(t_i), \quad 0 \leq t_i \leq t_{\max}$$

is a sequence of observations (**measurements, records**) of some real-valued signal $s(t)$. Each time series $s^{(m)}$ has its own number of observations $T^{(m)}$. Moreover, the sampling rate might be changing over time

$$t_i \neq i \cdot \frac{t_T - t_1}{T - 1},$$

or the ratio between different sampling rates might be non-rational. **How do we know??** In such cases as well as for time series with missing values we apply resampling procedure, which is described further.

A. Time series resampling

Let the time t be in continuous set \mathbb{R}_+^1 and the time series s be piece-wise constant. There are three possibilities to create such time series from a discrete-values one: 1) the constant goes after the sample $s(t)$, 2) before the sample, 3) in the neighborhood of the sample. See red, green and blue lines in the Figure 1a.

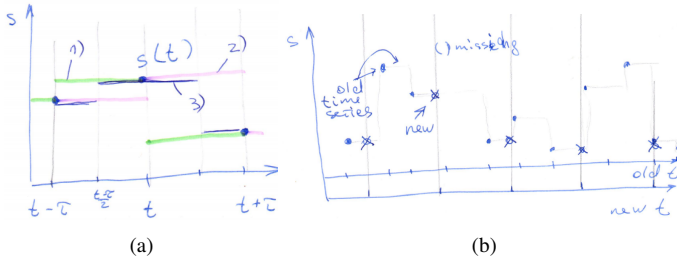


Figure 1: Piece-wise representation of a time series.

This assumptions helps introducing a new sampling rate and eliminates the problem of missing values, since the previous (next, current in the terms of Fig. 1a) value holds continuously until the following comes. The constant model could be developed into more complex one: a piece-wise linear, quadratic or cubic spline with its nodes in the time-ticks or over the time-ticks according to the following criterions: 1) NyquistShannon theorem, 2) Fisher-Neyman theorem. The following optimization problem returns the new sampling rate:

todo

This fixed rate is used to obtain a resampled time series with regular time-ticks.

a) *Selecting sampling rate: NyquistShannon sampling theorem.*: Suppose that the signal $s(t)$ is bandlimited with frequency f_b . According to the NyquistShannon sampling theorem, it is sufficient to sample the signal $s(t)$ with frequency

$$\frac{1}{\tau_s} = f_s > 2f_b \quad (1)$$

to be able to fully reconstruct the signal from its discretely sampled measurements $s(t_i) = s(i\tau_s)$. Alternatively, for $f_s > 2f_b$, Whittaker-Shannon interpolation

$$s(t) = \sum_{i=-\infty}^{\infty} s_i \text{sinc}\left(\frac{t - i\tau_s}{\tau_s}\right)$$

of the time series s yields perfect reconstruction of the signal $s(t)$. Sampling with $f_s < 2f_b$ causes distortions known as aliasing. In this case the time series have to be low-pass filtered to satisfy the Nyquist condition (1).

b) *FFT resampling.*: In signal processing a common way to change resolution of a signal is to use a combination of upsampling and decimation, implemented here via FFT transform and downsampling. Suppose that desired sampling rate f_s is fixed, that is, we would like to approximate $T_s = f_s t_{\max}$ uniformly sampled observations

$$s(i\tau_s), \quad i = 1, \dots, T_s, \quad \text{where } \tau_s = \frac{t_{\max}}{T_s - 1},$$

of the signal $s(t)$. Let $G = \{t_1, \dots, t_T\}$ and $G_s = \{\tau_s, \dots, \tau_s T_s\}$ denote the current and the desired grids. The first step to resampling $s(G) \rightarrow s(G_s)$ is the piecewise approximation \hat{s} of time series s at equidistant time points G_s (see Fig. 1a and Fig. 1b)

$$\hat{s}_i = s(t_i), \quad t_i = \max_{t \in G \text{ and } t \leq i\tau_s} t. \quad (2)$$

To increase the smoothness of piecewise approximation \hat{s} we apply low pass DFT filtering to \hat{s} , so that the reconstructed time series are bandlimited with $f_b < f_s/2$ and then down-sample the output to G_s .

On selecting f_s and f_b . Earlier we assumed that the desired sampling rate f_s is fixed. In fact, the new grid G_s is defined by the number of sampled observations T_s rather than the sampling frequency. Consider some special cases of unevenly sampled signals, where selecting T_s is straightforward.

1) The initial sampling rate is approximately even, but distortions are possible:

$$t_i = i \cdot \tau + \delta_i, \quad |\delta_i| < \frac{\tau}{2}.$$

In this case the number T_s of resampled observations equals the initial number of observations T .

2) The sampling rate is even, but some values are missing:

$$|t_{i+1} - t_i| = n\tau, n \in \mathbb{N}.$$

Here $\tau_s = \tau$ and missing values are the only ones that one needs to approximate.

3) Time series s comprises a finite number of intervals s_k , each sampled from $s(t)$ at fixed sampling rate:

$$s = \left[s(\tau_1), \dots, s(T_1\tau_1), s(T_1\tau_1 + \tau_2), \dots, s\left(\sum_k T_k\tau_k\right) \right],$$

where $\sum_k T_k = T$. Here we select the maximum sampling rate $f_s = \max_k \frac{1}{\tau_k}$ and upsample the rest time series, using piecewise constant approximation.

Another criterion for f_b . The cutoff frequency f_b is set so that fixed ratio $1 - \alpha$ of the power spectrum density is preserved:

$$\sum_{N f_b < k \leq N-1} |X_k|^2 < \alpha \sum_{k=0}^{N-1} |X_k|^2, \quad (3)$$

where X_k are the FFT coefficients of $s(iT)$. **The problem here is that we cannot estimate f_b until the time series are evenly sampled, which means that f_s has already been set and we only need to set f_b to $f_s/2$.**

c) *Selecting sampling rate: Fisher-Neyman resampling criterion.*: TODO

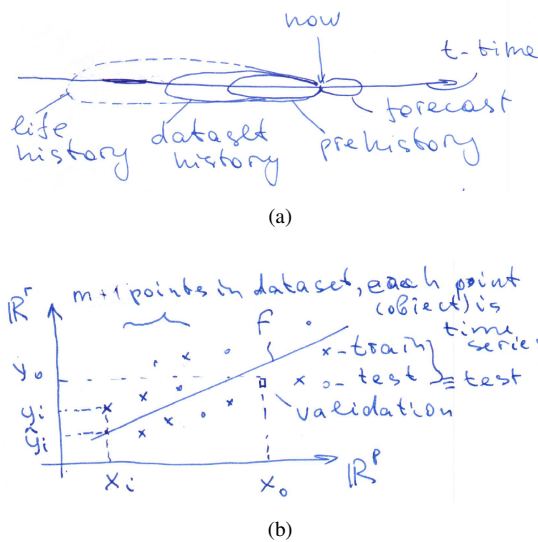


Figure 2: Forecasting (a) as regression problem (b).

Algorithm 1: FFT rescaling procedure.

TimeSeriesResampling()

begin

Data: time series $s = s(G)$, sampled at $G = \{t_1, \dots, t_T\}$.

Parameters: Upsampling rate f_s .

Result: resampled time series $\hat{s} = [\hat{s}(0), \dots, \hat{s}(t_{T_s})]$.

Form the new grid $G_s = \{0, \dots, T_s \cdot \tau_s\}$

Use piecewise approximation (2) to upsample time series: $\hat{s} = \uparrow_{G_s} s$

Apply low pass filtering to upsampled time series \hat{s} :

$\hat{s}^b = \text{LowPassFFTFiltering}(\hat{s}, \alpha)$

If necessary, downsample \hat{s}_b to G_s .

LowPassFFTFiltering()

begin

Data: Time series, s . **Parameters:** tolerance to spectrum reduction level α .

Result: Filtered time series s_b .

Zero-pad s , so that $|s| = 2^N$, where $N = \lceil \log_2(|s|) \rceil$

Find FFT coefficients a_j, b_j for $j = 1, \dots, N$ for s

Set f_b according to (3)

Set $a_j = 0, b_j = 0$, for $2\pi w_j > f_b$

Reconstruct the time series, using inverse FFT.

B. Design matrix

Given the set of properly resampled time series $\mathcal{D} = \{s^{(m)}\}, m = 1, \dots, M$, the task is to obtain forecasts $\hat{s}^{(m)}(t_i)$, $\Delta t_r < t_i \leq T_{\max} + \Delta t_r$ for each time series $s^{(m)}$ (Fig. 2a).

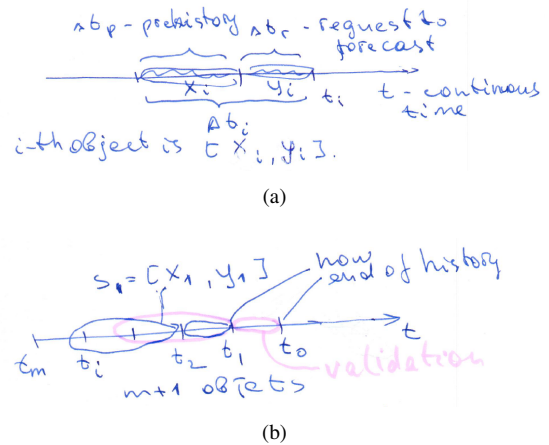


Figure 3: Draw an object from time series history.

To reformulate the forecasting task into regression task, form an object set at a set of the vectors $\{\mathbf{x}^*\}$, where each vector $\mathbf{x}^* = [\mathbf{y}|\mathbf{x}]$ collects all the time series over the time period Δt_p (Fig. 3a), which stands for the local *prehistory*. The vector \mathbf{x}^* includes samples from previous history of any time series $s^{(m)}$ as well as any derivatives ϕ , which are called generated features.

The design matrix \mathbf{X}^* for the multiscale autoregressive problem statement is constructed as follows (Fig. 3b). Let $s_i^{(m)}$ denote the i -th segment of the time series $s^{(m)}$

$$[\mathbf{x}_i^{(m)}|\mathbf{y}_i^{(m)}] = \underbrace{s^{(m)}(t_i - \Delta t_r - \Delta t_p), \dots, s^{(m)}(t_i - \Delta t_r)}_{\mathbf{x}_i^{(m)}}, \underbrace{s^{(m)}(t_i - \Delta t_r), \dots, s^{(m)}(t_i)}_{\mathbf{y}_i^{(m)}}, \quad (4)$$

where $s^{(m)}(t)$ is an element of time series $s^{(m)}$. To construct the design matrix, select t_i , $i = 1, \dots, q$ from $G = \{t_1, \dots, t_T\}$ such that segments $s_i = [\mathbf{x}_i|\mathbf{y}_i]$ cover time series s without intersection in target parts \mathbf{y}_i :

$$|t_{i+1} - t_i| > \Delta t_r. \quad (5)$$

Following (4) and (5), extract segments $[\mathbf{x}_i^{(m)}|\mathbf{y}_i^{(m)}]$, $i = 1, \dots, q$ from all time series $s^{(m)} \in \mathfrak{D}$ and form the matrix

$$\mathbf{X}^* = \left[\begin{array}{c|c} \mathbf{X} & \mathbf{Y} \\ \hline \mathbf{x} & \mathbf{y} \end{array} \right] = \left[\begin{array}{ccc|ccc} \mathbf{x}_1^{(1)} & \dots & \mathbf{x}_1^{(M)} & \mathbf{y}_1^{(1)} & \dots & \mathbf{y}_1^{(M)} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots \\ \mathbf{x}_q^{(1)} & \dots & \mathbf{x}_q^{(M)} & \mathbf{y}_q^{(1)} & \dots & \mathbf{y}_q^{(M)} \\ \hline \mathbf{x}^{(1)} & \dots & \mathbf{x}^{(M)} & \mathbf{y}^{(1)} & \dots & \mathbf{y}^{(M)} \end{array} \right].$$

Denote a row from the pair \mathbf{Y}, \mathbf{X} as \mathbf{y}, \mathbf{x} and call these vectors the target and the features.

a) *Feature selection and multimodelling.*: As one must forecast all elements from the target \mathbf{y} , only a few elements from the features \mathbf{x} are supposed to be informative in terms of the forecast quality. Denote the index set $\mathcal{J} = \{1, \dots, n\}$ and select the subset of the indexes $\mathcal{A} \in \mathcal{J}$. Introduce the forecasting model

$$\hat{\mathbf{y}}_i = \sum_{k=1}^K \pi_{ik} \mathbf{f}_k(\mathbf{w}_{\mathcal{A}_k}, \mathbf{x}_{i,\mathcal{A}_k})$$

as some linear combination of K models and call it the *multimodel*. Each model \mathbf{f}_k has its parameters \mathbf{w}_k and selected

features $\mathbf{x}_{\mathcal{A}_k}$. The coefficient π_{ik} set a vector \mathbf{x}_i in correspondence to the model \mathbf{f}_k , so that

$$\sum_{k=1}^K \pi_{ik} = 1 \quad \text{for } i \in \mathcal{I} = \{1, \dots, m\}$$

with two options are to be considered: $\pi \in \{0, 1\}$ and $\pi \in [0, 1]$. Let the forecasting error be

$$S = \sum_{i \in \mathcal{B}_0} \|\hat{\mathbf{y}}_i - \mathbf{y}_i\|_1,$$

where the set of object indexes \mathcal{I} is splitted to the test set \mathcal{B}_0 and the train sets,

$$\mathcal{I} = \mathcal{B}_0 \sqcup_{k=1}^K \mathcal{B}_k.$$

State the forecasting problem as a problem to minimize the error function S given models $\mathbf{f}_1, \dots, \mathbf{f}_K$ by optimizing matrix $\Pi = [\pi_{ik}]$, finite sets $\mathcal{A}_1, \dots, \mathcal{A}_K$ and model parameters $\mathbf{w}_1, \dots, \mathbf{w}_K$ on the sample set with indexes $\mathcal{I} \setminus \mathcal{B}_0$.

C. Special case of the problem

A special case of the problem is an early warning forecasting. There is a special time series \bar{s} with its elements $\bar{s} \in \{0, 1\}$. Here zero is interpreted as a *normal state* of the system and one means the system goes from normal to the *abnormal* state without return over time t . The problem is to maximize the lapse of the time segment

$$\|\Delta t_r\| \rightarrow \max,$$

where the vector

$$[\mathbf{x}_0, \mathbf{y}_0] = [0, \dots, 0, 0, \dots, 0, 1],$$

which means the system was in the normal state before it changes. Since the quality Q of forecasting time series \bar{s} depends on $\|\Delta t_r\|$ (the letter time lapse before the warning the higher the forecasting quality [ref]) the minimum level of quality must be set. Let the minimum forecasting quality be

$$Q\{(\hat{\mathbf{y}}_i, \bar{\mathbf{y}}_i) \mid i \in \mathcal{B}_0\} = \text{AUC} = Q_{\text{req}}.$$

a) *Forecast analysis.*: We consider the forecast testing procedure, given by the algorithm 2. Since our ultimate goal is to construct a forecasting model \mathbf{f} be able to obtain forecasts $\hat{\mathbf{y}}_i$ at any given time $t \leq t_i + \Delta t_r$ we have to imitate this setting, using the so-called retrospective forecast 4 (**rolling forecasts, walking ahead predictions**). Here we conceal most recent historical samples \mathbf{x}_0^* and make predictions as if they were unknown. Then the quality of the model \mathbf{f} is evaluated according to its performance on these concealed samples.

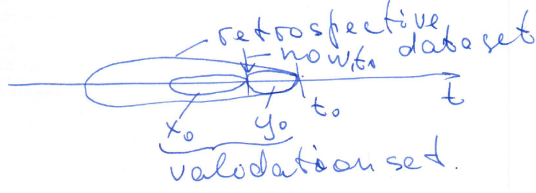


Figure 4: **Retrospective forecast includes most recent samples in data set.**

b) *Forecasting errors.*: Hyndman [27] divides forecasting errors into four types:

- scale-dependent metrics, such as mean absolute error

$$MAE = \frac{1}{r} \sum_{i=1}^r |\varepsilon_i|,$$

- percentage-error metrics such as the mean absolute percent error

$$MAPE = \frac{1}{r} \sum_{i=1}^r \frac{|\varepsilon_i|}{|y_0(i)|},$$

or symmetric MAPE

$$sMAPE = \frac{1}{r} \sum_{i=1}^r \frac{2|\varepsilon_i|}{|\hat{y}_0(i) + y_0(i)|},$$

- relative-error metrics, measure the average ratio of the errors from a designed method to the errors ε^* of a benchmark method

$$MRAE = \frac{1}{r} \sum_{i=1}^r \frac{|\varepsilon_i|}{\varepsilon_i^*},$$

- and scale-free error metrics, which express each error as a ratio to an average error from a martingal forecast:

$$MASE = \frac{n-1}{r} \frac{\sum_{i=1}^r |\varepsilon_i|}{\sum_{j=2}^n |x(j) - x(j-1)|}.$$

c) *Strijov's version of testing procedure*: To test consequent series of forecasts introduce the rolling forecasting procedure. It computes forecasts over sequence of time segments. The testing procedure counts K segments. Each segment starts at time-point t_k . It has a given length Δt_r . It varies from one time-tick to the length of the forecast horizon, which estimated before the experiment. To construct the design matrices for each time-point t_k split time-scale into K segments Δt_r up to the end. For each time-point t_k do the following:

- 1) construct the validation vector $\mathbf{x}_{val,k}^*$ for time series of the length Δt_r as the first row of the design matrix \mathbf{Z} ,

Algorithm 2: Train-test split.

ComputeForecastingErrors()

begin

Data: $\mathbf{X}^* \in \mathbb{R}^{M \times (\Delta t_r + \Delta t_p)}$. Parameters: sample size m , train to test ratio α .

Result: Forecasting quality: root-mean-squared error.

while $n \leq M - m$: **do**

define, $\mathbf{X}_n^* = [\mathbf{x}_n^*, \dots, \mathbf{x}_{m+n-1}^*]^\top$

$\mathbf{X}_{train}, \mathbf{X}_{test}, \mathbf{X}_{val} = TrainTestSplit(\mathbf{X}_n^*, \alpha)$

train forecasting model $\mathbf{f}(\mathbf{x}, \hat{\mathbf{w}}_n)$, using \mathbf{X}_{train} and \mathbf{X}_{test}

obtain vector of residuals $\varepsilon = [\varepsilon_T, \dots, \varepsilon_{T-\Delta t_r+1}]$ with respect to \mathbf{X}_{val}

compute forecasting quality:

$$MAPE(n) = \sqrt{\frac{1}{\Delta t_r} \sum_{t=0}^{\Delta t_r} \varepsilon_{T-t}^2};$$

$n = n + 1$

(or any adequate error function from III-C0b)

Average MAPE (other error) by data splits.

TrainTestSplit()

begin

Data: Object-feature matrix $\mathbf{X}^* \in \mathbb{R}^{m \times (\Delta t_r + \Delta t_p)}$.

Train to test ratio α .

Result: Train, test, validation matrices $\mathbf{X}_{train}^*, \mathbf{X}_{test}^*, \mathbf{X}_{val}^*$.

Set train set and test set sizes:

$$m_{train} = \lfloor \alpha(m-1) \rfloor$$

$$m_{test} = m - 1 - m_{train}$$

Decompose matrix \mathbf{X}^* into train, test, validation

matrices $\mathbf{X}_{train}^*, \mathbf{X}_{test}^*, \mathbf{X}_{val}^*$:

$$\mathbf{X}_{train}^* = \begin{bmatrix} \mathbf{x}_{val}^* \in \mathbb{R}^{1 \times (\Delta t_r + \Delta t_p)} \\ \mathbf{X}_{m_{test}}^* \in \mathbb{R}^{m_{test} \times (\Delta t_r + \Delta t_p)} \\ \mathbf{X}_{m_{train}}^* \in \mathbb{R}^{m_{train} \times (\Delta t_r + \Delta t_p)} \end{bmatrix} = \begin{bmatrix} \mathbf{y}_{val} & \mathbf{x}_{val} \\ \mathbf{Y}_{m_{test}} & \mathbf{X}_{m_{test}} \\ \mathbf{Y}_{m_{train}} & \mathbf{X}_{m_{train}} \end{bmatrix}$$

- 2) construct the rest rows of the design matrix \mathbf{Z} for the time

after t_k and present it as

$$\mathbf{Z} = \begin{bmatrix} \dots & \dots \\ \mathbf{x}_{\text{val},k} & \mathbf{y}_{\text{val},k} \\ 1 \times n & 1 \times r \\ \mathbf{x}_{\text{train},k} & \mathbf{y}_{\text{train},k} \\ m_{\text{min}} \times n & m_{\text{min}} \times r \\ \dots & \dots \end{bmatrix},$$

- 3) optimize model parameters (\mathbf{w}, β) using $\mathbf{X}_{\text{train},k}, \mathbf{Y}_{\text{train},k}$, (run additional cross-validation procedure on there rows to select the optimal model structure, if necessary),
- 4) use $\mathbf{x}_{\text{val},k}, \mathbf{y}_{\text{val},k}$ to compute the forecast $(\mathbf{x}_{\text{val}}\mathbf{w}, \beta)$ and its residues $(\mathbf{f} - \mathbf{y})$,
- 5) compute MAPE (??).

IV. FEATURE GENERATION

List of procedures for constructing the feature and the object sets will be placed here. Discussion point: vector \mathbf{y} **remains always unchanged**.

The feature set $\mathcal{J} = \bigcup_k \mathcal{A}_k$ includes

- 1) the local history of all time series themselves,
- 2) transformations (non-parametric and parametric) of local history,
- 3) parameters of the local models,
- 4) distances to the centroids of local clusters.

The object set $\mathcal{I} = \bigcup_k \mathcal{B}_k$ includes

- 1) the local history,
- 2) parametric local models and their residuals (including ones from previous iterations),
- 3) DTW-shifted local history as a local forecasting procedure,
- 4) aggregated subsets of time series.

Denote the generated feature vector as ϕ . This vector consists of concatenated row-vectors $\phi = [\phi^{(1)}, \dots, \phi^{(M)}]$, which corresponds to time series local histories $\mathbf{s} = [\mathbf{s}^{(1)}, \dots, \mathbf{s}^{(M)}]$, modified with set of transformations \mathcal{G} . The elements $g : \mathbf{s} \rightarrow \phi$ of this set are listed below.

A. Transformations of local history

The tables II, III, IV, V, VI list the time series transformation functions. There are non-parametric and parametric procedures to generate features. For the parametric functions $g = g(\mathbf{b}, \mathbf{s})$ the default values of the parameters \mathbf{b} are assigned empirically.

The parametric procedure request two optimization problem statements of the model parameters \mathbf{w} and the primitive function parameters \mathbf{b} . The first one fixes the vector $\hat{\mathbf{b}}$, collected over all the primitive functions $\{g\}$, which generate features ϕ :

$$\hat{\mathbf{w}} = \arg \min S(\mathbf{w} | \mathbf{f}(\mathbf{w}, \mathbf{x}), \mathbf{y}), \quad \text{where } [\mathbf{y}, \mathbf{x}] = \phi(\hat{\mathbf{b}}, \mathbf{s}).$$

The second one optimizes the transformation parameters $\hat{\mathbf{b}}$ given obtained model parameters \mathbf{w}

$$\hat{\mathbf{b}} = \arg \min S(\mathbf{b} | \mathbf{f}(\hat{\mathbf{w}}, \mathbf{x}), \mathbf{y}).$$

This procedure repeats two problems until vectors $\hat{\mathbf{w}}, \hat{\mathbf{b}}$ converge. **The initial values of vector \mathbf{b} (are shown in table ??).** Due to the various origins of the time series and their transformations the residual vector should be normalized:

$$\epsilon = \frac{\hat{\mathbf{y}} - \mathbf{y}}{|\mathbf{y}| \cdot \|\mathbf{y}\|_2^1}.$$

It (transformation? normalization) does not change the number elements in the vectors, $|\phi| = |\mathbf{s}|$.

B. Convolutions, statistics and parameters of local history

The listed feature generation functions convolves time series, so they reduce the dimensionality $|\phi = \mathbf{g}(\mathbf{s})| < |\mathbf{s}|$.

C. Parameters of local history forecast

For the time series \mathbf{s} construct the Hankel matrix with a period k and shift p , so that for $\mathbf{s} = [s_1, \dots, s_T]$ the matrix

$$\mathbf{H}^* = \begin{bmatrix} s_T & \dots & s_{T-k+1} \\ \vdots & \ddots & \vdots \\ s_{k+p} & \dots & s_{1+p} \\ s_k & \dots & s_1 \end{bmatrix}, \quad \text{where } 1 \geq p \geq k.$$

Reconstruct the regression to the first column of the matrix $\mathbf{H}^* = [\mathbf{h}, \mathbf{H}]$ and denote its least square parameters as the feature vector

$$\phi^{(m)} = \arg \min \|\mathbf{h} - \mathbf{H}\phi\|_2^2.$$

For the time series $\mathbf{s}^{(m)}$, $m = 1, \dots, M$ use the parameters $\phi^{(m)}$ as the features.

D. Distances to centroids of local clusters

This procedure applies the kernel trick to the time series. For given local history time series $\mathbf{x}_i^{(m)}$, $m = 1, \dots, M$ compute k -means centroids $\mathbf{c}_p^{(m)}$, $p = 1, \dots, P$. With the selected k -means distance function ρ construct the feature vector

$$\phi_i^{(m)} = [\rho(\mathbf{c}_1^{(m)}, \mathbf{s}_i^{(m)}), \dots, \rho(\mathbf{c}_P^{(m)}, \mathbf{s}_i^{(m)})] \in \mathbb{R}_+^P.$$

This k -means of another clustering procedure may use internal parameters, so that there are no parameters to be included to the feature vector or to the forecasting model.

Table II: Must-try functions.

Formula	Output dimension	# of arguments	# of parameters
\sqrt{x}	1	1	0
$x\sqrt{x}$	1	1	0
$\arctan x$	1	1	0
$\ln x$	1	1	0
$x \ln x$	1	1	0

V. FEATURE SELECTION

TODO

VI. MIXTURE MODELS

Let $D = (X, \mathbf{y})$ denote the data, where $X = [\mathbf{x}_1^\top, \dots, \mathbf{x}_i^\top, \dots, \mathbf{x}_m^\top]^\top$, denotes the inputs $\mathbf{x}_i \in \mathbb{R}^n$, \mathbf{y} denotes the targets $y_i \in Y$. The task is to estimate y_i , given \mathbf{x}_i . Assuming linear model f with gaussian noise

$$y = f(\mathbf{x}, \mathbf{w}) + \varepsilon, f(\mathbf{x}, \mathbf{w}) = \mathbf{w}^\top \mathbf{x}, \varepsilon \sim \mathcal{N}(0, \beta) \Rightarrow y \sim \mathcal{N}(\mathbf{w}^\top \mathbf{x}, \beta),$$

obtain the maximum likelihood estimate

$$\hat{y} = \hat{\mathbf{w}}^\top \mathbf{x}, \hat{\mathbf{w}} = \arg \max_{\mathbf{w}} \frac{1}{2\beta} \sum_{i=1}^m (y_i - \mathbf{w}^\top \mathbf{x}_i)^2$$

for the output.

A. EM-algorithm for mixture models

Assume the target variable \mathbf{y} is generated by one of K linear models $f_k(\mathbf{x}, \mathbf{w}_k)$. Let the distribution of the target variable \mathbf{y} be a mixture of normal distributions

$$p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{y}|\mathbf{w}_k^\top \mathbf{x}, \beta) = \quad (6)$$

$$\sum_{k=1}^K \frac{1}{(2\pi\beta_k)^{n/2}} \exp \left(-\frac{1}{2\beta_k} (\mathbf{y} - \mathbf{w}_k^\top \mathbf{x})^\top (\mathbf{y} - \mathbf{w}_k^\top \mathbf{x}) \right).$$

Here $\boldsymbol{\theta}$ denotes the concatenated vector of parameters:

$$\boldsymbol{\theta} = [\mathbf{w}_1, \dots, \mathbf{w}_K, \boldsymbol{\pi}, \beta]^\top,$$

where $\boldsymbol{\pi} = [\pi_1, \dots, \pi_K]$ are weights of the models, and $\mathbf{B} = \beta \mathbf{I}_m$ is the covariance matrix for \mathbf{y} .

a) *Parameter estimation.*: The goal is to find parameters vector $\hat{\boldsymbol{\theta}}$ which optimizes loglikelihood function for given data set D

$$\hat{\boldsymbol{\theta}} = \arg \max_{\boldsymbol{\theta}} \ln p(\mathbf{y}|\boldsymbol{\theta}), \ln p(\mathbf{y}|\boldsymbol{\theta}) = \sum_{i=1}^m \ln \left(\sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{y}|\mathbf{w}_k^\top \mathbf{x}_i, \beta) \right). \quad (7)$$

To obtain maximum likelihood estimates (7) for parameter $\boldsymbol{\theta}$ of the model (6), let us introduce hidden indicator variables

$$Z = [\mathbf{z}_1, \dots, \mathbf{z}_m], \quad z_{ik} \in \{0, 1\},$$

such that

$$z_{ik} = 1 \Leftrightarrow y_i \sim \mathcal{N}(\mathbf{w}_k^\top \mathbf{x}_i, \beta).$$

Then the loglikelihood function $p(\mathbf{y}, Z|X, \boldsymbol{\theta})$ takes the form

$$p(\mathbf{y}|X, Z, \boldsymbol{\theta}) = \sum_{i=1}^m \sum_{k=1}^K z_{ik} (\ln \pi_k + \ln \mathcal{N}(y_i|\mathbf{w}_k^\top \mathbf{x}_i, \beta)) = \\ = \sum_{i=1}^m \sum_{k=1}^K z_{ik} \left(\ln \pi_k - \frac{1}{2\beta} (y_i - \mathbf{w}_k^\top \mathbf{x}_i)^2 + \frac{n \ln \beta}{2} + \text{const} \right).$$

Since $p(\mathbf{y}, Z|X, \boldsymbol{\theta})$ depends on random variables z_{ik} , instead of $p(\mathbf{y}|X, \boldsymbol{\theta})$ maximize the expected loglikelihood of the observed data D :

$$\mathbb{E}_Z[p(\mathbf{y}, Z|X, \boldsymbol{\theta})] =$$

$$\sum_{i=1}^m \sum_{k=1}^K \gamma_{ik} \left(\ln \pi_k - \frac{1}{2\beta} (y_i - \mathbf{w}_k^\top \mathbf{x}_i)^2 + \frac{n \ln \beta}{2} \right),$$

where $\gamma_{ik} = \mathbb{E}[z_{ik}|\mathbf{y}, X]$.

Finally, apply Expectation-Maximization algorithm to maximize $\mathbb{E}_Z[p(\mathbf{y}, Z|X, \boldsymbol{\theta})]$ updating parameters estimates $\boldsymbol{\theta}^{(r)}$ in two iterative steps.

E-step: obtain $\mathbb{E}(Z)$. Let $\Gamma = [\gamma_{ik}]$ be a matrix of posterior probabilities that i -th sample is generated by k -th model. Using Bayesian rule, obtain

$$\gamma_{ik}^{(r+1)} = \mathbb{E}(z_{ik}) = p(k|\mathbf{x}_i, \boldsymbol{\theta}^{(r)}) = \quad (8)$$

$$\frac{\pi_k \mathcal{N}(y_i|\mathbf{x}_i^\top \mathbf{w}_k^{(r)}, \beta^{(r)})}{\sum_{k'=1}^K \pi_{k'} \mathcal{N}(y_i|\mathbf{x}_i^\top \mathbf{w}_{k'}^{(r)}, \beta^{(r)})}.$$

Define expectations of joint loglikelihood $\ln p(\mathbf{y}, Z|X, \boldsymbol{\theta})$ with respect to the posteriors distribution $p(Z|\mathbf{y}, \boldsymbol{\theta})$

$$Q^{(r)}(\boldsymbol{\theta}) = \mathbb{E}_Z(\ln p(\mathbf{y}, Z|\boldsymbol{\theta})) = \sum_{i=1}^m \sum_{k=1}^K \gamma_{ik}^{(r+1)} \quad (9)$$

$$\left(\ln \pi_k^{(r)} + \ln \mathcal{N}(y_i|\mathbf{x}_i^\top \mathbf{w}_k^{(r)}, \beta^{(r)}) \right).$$

Table III: List of elementary functions.

Function name	Formula	Output dimension	# of arguments	# of parameters
Add constant	$x + w$	1	1	1
Quadratic	$w_2 x^2 + w_1 x + w_0$	1	1	3
Cubic	$w_3 x^3 + w_2 x^2 + w_1 x + w_0$	1	1	4
Logarithmic sigmoid	$1/(w_0 + \exp(-w_1 x))$	1	1	2
Exponent	$\exp x$	1	1	0
Normal	$\frac{1}{w_1 \sqrt{2\pi}} \exp\left(\frac{(x-w_2)^2}{2w_1^2}\right)$	1	1	2
Multiply by constant	$x \cdot w$	1	1	1
Monomial	$w_1 x^{w_2}$	1	1	2
Weibull-2	$w_1 w_2 x^{w_2-1} \exp -w_1 x^{w_2}$	1	1	2
Weibull-3	$w_1 w_2 x^{w_2-1} \exp -w_1 (x - w_3)^{w_2}$	1	1	3

Table IV: Monotone functions.

By growth rate					
Function name	Formula	Output dimension	# of arguments	# of parameters	Constraints
Linear	$w_1 x + w_0$	1	1	2	
Exponential rate	$\exp(w_1 x + w_0)$	1	1	2	$w_1 > 0$
Polynomial rate	$\exp(w_1 \ln x + w_0)$	1	1	2	$w_1 > 1$
Sublinear polynomial rate	$\exp(w_1 \ln x + w_0)$	1	1	2	$0 < w_1 < 1$
Logarithmic rate	$w_1 \ln x + w_0$	1	1	2	$w_1 > 0$
Slow convergence	$w_0 + w_1/x$	1	1	2	$w_1 \neq 0$
Fast convergence	$w_0 + w_1 \cdot \exp(-x)$	1	1	2	$w_1 \neq 0$
Other					
Soft ReLu	$\ln(1 + e^x)$	1	1	0	
Sigmoid	$1/(w_0 + \exp(-w_1 x))$	1	1	2	$w_1 > 0$
Nonparametric log-sigmoid	$1/(1 + \exp(-x))$	1	1	0	
Hiberbolic tangent	$\tanh(x)$	1	1	0	
softsign	$\frac{ x }{1+ x }$	1	1	0	

Table V: Multivariate.

Bivariate				
Plus	$x_1 + x_2$	1	2	0
Minus	$x_1 - x_2$	1	2	0
Product	$x_1 \cdot x_2$	1	2	0
Division	$\frac{x_1}{x_2}$	1	2	0
	$x_1 \sqrt{x_2}$	1	2	0
	$x_1 \ln x_2$	1	2	0
Multivariate				
Sum of products	$\sum_{i,j} x_i x_j$	1	$n \geq 2$	0
Sum of products	$\sum_{i,j,k} x_i x_j x_k$	1	$n \geq 3$	0
Sum of Gaussians	$\sum_{j=1}^n a_j \exp(-\frac{(x_j - b_j)^2}{c_j})$	1	n	$3n$
Polynomial	$\sum_{j=0}^n a_j x^j$	1	1	n
Rational polynomial	$\frac{\sum_{j=0}^n a_j x^j}{x^m + \sum_{j=0}^{m-1} b_j x^j}$	1	1	$n + m + 1$

M-step: update parameters θ , maximizing $Q^{(r)}(\theta)$. Maximize function $Q^{(r)}(\theta)$ with respect to θ with $\Gamma^{(r+1)}$ fixed. First, optimize π_k , which is constrained as $\sum_{k=1}^K \pi_k = 1$.

Using Lagrange multipliers, obtain the following estimation

$$\pi_k^{(r+1)} = \frac{1}{n} \sum_{i=1}^m \gamma_{ik}^{(r+1)}.$$

Table VI: Data statistics.

sum	$\sum_i x_i$	1	m	0
mean	$(\sum_i x_i)/m$	1	m	0
min	$\min_i x_i$	1	m	0
max	$\max_i x_i$	1	m	0
std	$\frac{1}{m-1} \sqrt{\sum_i (x_i - \text{mean}(x))^2}$	1	m	0
hist	$\sum_i [X_{j-1} < x_i \leq X_j]$	n	m	$n - 1$
conv	$\sum_j x_{i-j} w_j$	1	$m - n + 1$	$n \leq m$
FFT coefficients		n	m	1

Next, maximize $Q^{(r)}$ with respect to \mathbf{w}_k for k -th model. With π_k fixed maximizing (9) is equivalent to

$$\mathbf{w}_k^{(r+1)} = \arg \max_{\mathbf{w}_k} \sum_{i=1}^m -\gamma_{ik}^{(r+1)} (y_i - \mathbf{w}_k^T \mathbf{x}_i)^2,$$

$$\beta_k^{(r)} = \arg \max_{\beta} \sum_{i=1}^m \gamma_{ik}^{(r+1)} \left(n \ln \beta - \frac{1}{\beta} (y_i - \mathbf{x}_i^T \mathbf{w}_k^{(r+1)})^2 \right).$$

B. Mixture of experts

Suppose that each model $f(\mathbf{x}, \mathbf{w}_k)$ generates a sample (\mathbf{x}, y) with some probability $p(k|\mathbf{x}, \mathbf{w})$. Then the following factorization holds

$$p(y|\mathbf{x}, \boldsymbol{\theta}) = \sum_{k=1}^K p(y, k|\mathbf{x}, \boldsymbol{\theta}) = \sum_{k=1}^K p(k|\mathbf{x}, \boldsymbol{\theta}) p(y|k, \mathbf{x}, \boldsymbol{\theta})$$

for $p(y|\mathbf{x}, \boldsymbol{\theta})$. Here $p(k|\mathbf{x}, \boldsymbol{\theta})$ correspond to weight parameters π_k in (6) dependent on the inputs \mathbf{x} . Assuming normal linear models $f(\mathbf{x}, \mathbf{w}_k)$ or, equivalently, normal distributions $p(y|\mathbf{x}, \mathbf{w}_k) = \mathcal{N}(y|\mathbf{w}_k^T \mathbf{x}, \beta)$, obtain

$$p(y|\mathbf{x}, \boldsymbol{\theta}) = \sum_{k=1}^K \pi_k(\mathbf{x}, \mathbf{v}_k) \mathcal{N}(y|\mathbf{w}_k^T \mathbf{x}, \beta), \quad (10)$$

where

$$\pi_k(\mathbf{x}, \mathbf{v}_k) = \frac{\exp(\mathbf{v}_k^T \mathbf{x})}{\sum_{k'=1}^K \exp(\mathbf{v}_{k'}^T \mathbf{x})}.$$

The difference between mixture of experts model (10) and mixture model (6) is that model weights π_k depend on inputs \mathbf{x} in mixture of experts. Similarly, EM-procedure for mixture of experts differs from EM-procedure for mixture models in the way γ_{ik} are optimized in M-step.

C. Distance between two models of N time series

Introduce a distance function $\rho(f_k, f_l)$ between two models. Use the Jensen-Shannon divergence; $\rho_{kl} \in [0, 1]$ is a metric:

$$\rho(p_k \| p_l) = 2^{-1} D_{\text{KL}}(p_k \| p') + 2^{-1} D_{\text{KL}}(p' \| p_l),$$

Algorithm 3: EM-algorithm for mixture of experts.

begin

Data: (\mathbf{x}_i, y_i) , $i = 1, \dots, m$. Parameters: number of experts K .

Result: Parameters $\boldsymbol{\theta}$ of the model (10).

Initialize $[\mathbf{w}, \beta, \mathbf{v}] \equiv \boldsymbol{\theta} = \boldsymbol{\theta}^{(0)}$, $r = 0$

while $\boldsymbol{\theta}$ keeps changing **do**

E step: compute hidden variables $\gamma_{ik}^{(r+1)}$, the expectation of the indicator variables, using (8)

M step: find new parameter estimates

$\mathbf{v}_k^{(r+1)} = \arg \max_{\mathbf{v}} Q_k^{(r), \mathbf{v}}(\mathbf{v})$, where

$$Q_k^{(r), \mathbf{v}}(\mathbf{v}) = \sum_{i=1}^m \gamma_{ik}^{(r+1)} \ln \pi_k(\mathbf{x}_i, \mathbf{v})$$

$\mathbf{w}_k^{(r+1)} = \arg \max_{\mathbf{w}_k} Q_k^{(r), \mathbf{w}}(\mathbf{w}_k)$, where

$$Q_k^{(r), \mathbf{w}}(\mathbf{w}_k) = \sum_{i=1}^m \gamma_{ik}^{(r+1)} (y_i - \mathbf{w}_k^T \mathbf{x}_i)^2,$$

$\beta_k^{(r+1)} = \arg \max_{\beta} Q_k^{(r), \beta}(\beta)$, where

$$Q_k^{(r), \beta}(\beta) = \left(n \ln \beta - \frac{1}{\beta} (y_i - \mathbf{x}_i^T \mathbf{w}_k^{(r+1)})^2 \right)$$

where $p' = 2^{-1}(p_k + p_l)$ and $p_k \stackrel{\text{def}}{=} (p(\mathbf{w}|D, A, B, f_k))$. The non-symmetric Kullback-Leibler divergence is

$$D_{\text{KL}}(p \| p') = \int_{\mathbf{w} \in \mathbb{W}} p'(\mathbf{w}) \ln \frac{p(\mathbf{w})}{p'(\mathbf{w})} d\mathbf{w}.$$

VII. COMPUTATIONAL EXPERIMENT

The goal of the experiment is to compare the following four approaches to the multiscale forecasting: 1) Bayesian mixture model approach, 2) random multimodel, 3) vector random decision forest and 4) vector adaboost. The last two algorithms are modifications of [\[link\]](#). The modifications are needed to produce the vector of multiscale time series as their outputs. The experiment is performed on 1) non-modified autoregression data and on 2) data with additionally generated features as it is described in the corresponding section.

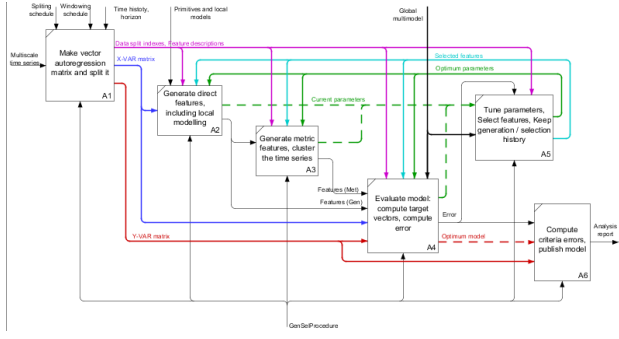


Figure 5: Multiscale forecasting pipeline.

VIII. APPENDIX: DISCRETE GENETIC ALGORITHM FOR FEATURE SELECTION (WILL BE CONVERTED TO BOOTSTRAP RANDOM LINEAR MULTIMODEL ALGORITHM)

- 1) There are set of binary vectors $\{a_1, \dots, a_P\}$, $a \in \{0, 1\}^n$;
- 2) get two vectors a_p, a_q , $p, q \in \{1, \dots, P\}$;
- 3) chose random number $\nu \in \{1, \dots, n-1\}$;
- 4) split both vectors and change their parts:

$$[a_{p,1}, \dots, a_{p,\nu}, a_{q,\nu+1}, \dots, a_{q,n}] \rightarrow a'_p,$$

$$[a_{q,1}, \dots, a_{q,\nu}, a_{p,\nu+1}, \dots, a_{p,n}] \rightarrow a'_q;$$

- 5) choose random numbers $\eta_1, \dots, \eta_Q \in \{1, \dots, n\}$;
- 6) invert positions η_1, \dots, η_Q of the vectors a'_p, a'_q ;
- 7) repeat items 2-6 $P/2$ times;
- 8) evaluate the obtained models.

Repeat R times; here P, Q, R are the parameters of the algorithm and n is the number of the corresponding model features.

IX. APPENDIX: MIXTURE MODELLING UNDER RANDOM BOOTSTRAPPED MODELS

Denote the indexes of objects as $\{1, \dots, i, \dots, m\} = \mathcal{I}$, the split $\mathcal{I} = \mathcal{B}_1 \sqcup \dots \sqcup \mathcal{B}_K$ and the indexes of features as $\{1, \dots, j, \dots, n\} = \mathcal{J}$, the active set $\mathcal{A}_k \subseteq \mathcal{J}$.

Let the regression model

$$f : (w, x) \mapsto y;$$

with the selected model of optimal structure

$$E(y_i | x) = W_{\mathcal{A}} x_i.$$

The multimodel f is a set of the models $f = \{f_k \mid k = 1, \dots, K\}$, such that for each k

$$E(y_{i \in \mathcal{B}_k} | x) = W_{\mathcal{A}_k} x_{i \in \mathcal{B}_k} \quad \text{with} \quad \mathcal{I} = \sqcup_{k=1}^K \mathcal{B}_k \ni i.$$

State the multimodel selection problem as follows. An optimal single model is

$$\hat{f}(w, x) = \arg \max_{\mathcal{A} \subseteq \mathcal{J}} \mathcal{E}(f(w_{\mathcal{A}}, x)),$$

where \mathcal{E} denotes the model evidence in coherent Bayesian inference. An optimal multilevel model is

$$\hat{f}(w_1, \dots, w_K, x) = \arg \max_{\sqcup_{k=1}^K \mathcal{B}_k = \mathcal{I}} \prod_{k=1}^K \mathcal{E}(f(w_k, x_{\mathcal{B}_k})).$$

~~The model difference must be statistically significant~~

$$\mathcal{F} \supset \hat{f} = \arg \max_{\mathcal{B}_1, \mathcal{B}_2 \subset \mathcal{B}} \rho(f_1, f_2)$$

~~given set of indices $\hat{\mathcal{A}}$, such that~~

$$\hat{\mathcal{A}} = \arg \max_{\mathcal{A} \subseteq \mathcal{J}} \mathcal{E}(f_1(w_{\mathcal{A}}, x^{\mathcal{B}_1})) \mathcal{E}(f_2(w'_{\mathcal{A}}, x^{\mathcal{B}_2})).$$

REFERENCES

- [1] M. D. Costa, C.-K. Peng, and A. L. Goldberger, "Multiscale analysis of heart rate dynamics: Entropy and time irreversibility measures," *Cardiovascular Engineering*, vol. 8, no. 2, pp. 88–93, 2008.
- [2] M. U. Ahmed, N. Rehman, D. Looney, T. M. Rutkowski, P. Kidmose, and D. P. Mandic, "Multivariate entropy analysis with data-driven scales," *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3901 – 3904, 2012.
- [3] P. Cortez, M. Rio, M. Rocha, and P. Sousa, "Multi-scale internet traffic forecasting using neural networks and time series methods," *Expert Systems*, vol. 29, no. 2, pp. 143–155, 2012.
- [4] M. A. R. Ferreira, D. M. Higdon, H. K. H. Lee, and M. West, "Multi-scale and hidden resolution time series models," *Bayesian Analysis*, vol. 1, no. 4, pp. 947–967, 2006.
- [5] Z. Cui, W. Chen, and Y. Chen, "Multi-scale convolutional neural networks for time series classification," *Computer Vision and Pattern Recognition*, 2016.
- [6] C. Aldrich and L. Auret, *Process Monitoring and Fault Diagnosis with Machine Learning Methods (Advances in Computer Vision and Pattern Recognition)*. Springer London, 2013, ch. Process Monitoring Using Multiscale Methods, pp. 341–369.
- [7] S.-D. Wu, C.-W. Wu, S.-G. Lin, C.-C. Wang, and K.-Y. Lee, "Time series analysis using composite multiscale entropy," *Entropy*, vol. 15, no. 3, pp. 1069–1084, 2013.
- [8] Y. Jiang, C.-K. Peng, and Y. Xu, "Hierarchical entropy analysis for biological signals," *Journal of Computational and Applied Mathematics*, vol. 236, p. 728742, 2011.
- [9] H. Chen, B. Vidakovic, , and D. Mavris, "Multiscale forecasting method using armax models," Georgia Institute of Technology, Tech. Rep., 2004.
- [10] U. Vespier, A. Knobbe, S. Nijssen, and J. Vanschoren, *Machine Learning and Knowledge Discovery in Databases*, ser. Lecture Notes in Computer Science, 2012, vol. 7524, ch. MDL-Based Analysis of Time Series at Multiple Time-Scales, pp. 371–386.
- [11] T. B. Trafalis and H. Ince, "Support vector machine for regression and applications to financial forecasting," in *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN2000)*, 2000, pp. 348–353.

- [12] R. Navarrete and D. Viswanath. (2015) Support vector regression, smooth splines, and time series prediction. [Online]. Available: arXiv:1511.00158v1
- [13] W. Hao and S. Yu, *Knowledge Enterprise: Intelligent Strategies in Product Design, Manufacturing, and Management. Proceedings of PRO-LAMAT 2006, IFIP TC5 International Conference, June 1517, 2006, Shanghai, China, 2006*, ch. Support Vector Regression for Financial Time Series Forecasting, pp. 825–830.
- [14] R. Yu, Y. Yang, L. Yang, G. Han, and O. A. Move, “Raga random forest approach for predicting air quality in urban sensing systems sensors 2016,” *Sensors*, vol. 16, no. 1, p. 86, 2016.
- [15] M. J. Kane, N. Price, M. Scotch, and P. Rabinowitz, “Comparison of arima and random forest time series models for prediction of avian influenza h5n1 outbreaks,” *BMC Bioinformatics*, vol. 15, no. 276, 2014.
- [16] E. Busseti, I. Osband, and S. Wong, “Compared kernalized regression and 3 types of nn using the data from kaggle competition global energy forecasting competition 2012 - load forecasting,” Stanford University, Tech. Rep., 2012.
- [17] G. W. Taylor and G. E. Hinton, “Factored conditional restricted boltzmann machines for modeling motion style,” *ICML '09 Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 1025–1032, 2009.
- [18] A. Criminisi, J. Shotton, and E. Konukoglu, *Decision Forests: A Unified Framework for Classification, Regression, Density Estimation, Manifold Learning and Semi-Supervised Learning*, ser. Foundations and Trends in Computer Graphics and Vision, 2011, vol. 7, no. 2-3, ch. Regression Forests, pp. 131–148.
- [19] Y. Bao, T. Xiong, and Z. Hu, “Multi-step-ahead time series prediction using multiple-output support vector regression,” *Neurocomputing*, vol. 129, pp. 482–493, 2014.
- [20] L. Zhang, W.-D. Zhou, P.-C. Chang, J.-W. Yang, and F.-Z. Li, “Iterated time series prediction with multiple support vector regression models,” *Neurocomputing*, vol. 99, no. 1, p. 411422, 2013.
- [21] F. Pérez-Cruz, G. Camps-Valls, E. Soria-Olivas, J. Pérez-Ruixo, A. Figueiras-Vidal, and A. Artés-Rodríguez, “Multi-dimensional function approximation and regression estimation,” *Artificial Neural Networks –ICANN*, pp. 796–796, 2002.
- [22] X. Qiu, Nanyang, L. Zhang, Y. Ren, P. N. Suganthan, and G. Amaratunga, “Browse conference publications 4 computational intelligence in ... help working with abstracts ensemble deep learning for regression and time series forecasting,” in *Computational Intelligence in Ensemble Learning (CIEL), 2014 IEEE Symposium on*, 2014.
- [23] A. Grover, A. Kapoor, and E. Horvitz, “A deep hybrid model for weather forecasting,” in *KDD '15 Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, pp. 379–386.
- [24] B. Zhu, “A novel multiscale ensemble carbon price prediction model integrating empirical mode decomposition, genetic algorithm and artificial neural network,” *Energies*, vol. 5, pp. 355–370, 2012.
- [25] Y. Bai, Z. Chen, J. Xie, and C. Li, “Daily reservoir inflow forecasting using multiscale deep feature learning with hybrid models,” *Journal of Hydrology*, vol. 532, pp. 193–206, 2015.
- [26] S. Ferrari, F. Bellocchio, V. Piuri, and N. A. Borghese, “Hierarchical approach for multiscale support vector regression,” *IEEE Transactions on Neural Networks Learning Systems*, vol. 23, no. 9, pp. 1448–1460, 2012.
- [27] R. J. Hyndman, “Another look at forecast-accuracy metrics for intermittent demand,” *Foresight*, no. 4, pp. 43–46, June 2006.
- [28] A. Ignatov and V. Strijov, “Human activity recognition using quasiperiodic time series collected from a single triaxial accelerometer,” *Multi-media Tools and Applications*, pp. 1–14, 2015.
- [29] T. Kuremoto, S. Kimura, K. Kobayashi, and M. Obayashi, *Emerging Intelligent Computing Technology and Applications*, 2012, ch. Time Series Forecasting Using Restricted Boltzmann Machine, pp. 17–22.
- [30] M. F. Matthias Scholz and J. Selbig., *Principal Manifolds for Data Visualization and Dimension Reduction*, ser. LNCSE. Springer Berlin Heidelberg, 2007, vol. 58, ch. Nonlinear principal component analysis: neural network models and applications, pp. 44–67.
- [31] A. Motrenko and V. Strijov, “Extracting fundamental periods to segment human motion time series,” 2016.
- [32] S. Yuksel, J. N. Wilson, and P. Gader, “Twenty years of mixture of experts,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 8, pp. 1177–1193, 2015.
- [33] [Online]. Available: <http://gdudek.el.pcz.pl/varia/stlf-data>
- [34] [Online]. Available: <http://hasc.jp/hc2010/HASC2010corpus/hasc2010corpus-en.html>
- [35] [Online]. Available: <http://www.neural-forecasting-competition.com/NN3/datasets.htm>