

Tensor decomposition and forecast for multidimensional time series

Semkin Kirill¹ and Strijov Vadim²

¹Moscow Institute of Physics and Technology, semkin.ki@phystech.edu

²Moscow Institute of Physics and Technology, strijov@ccas.ru

Abstract

Processing of multidimensional time series is associated with additional task of modeling interconnections between signals. Its ignorance may cause incorrectness of any method's outcomes. On the other hand, taking this dependencies into account makes models larger and less interpretable as well as series analysis more narrow. A non-parametric method based on tensor data representation and SSA approach is proposed in the paper. Then, series decomposition technique and forecast type and formula were derived. Finally, elaborated theory was applied to electricity consumption and weather data. Obtained results were compared with mSSA, VAR, RNN models.

Key words: time series, decomposition, forecast, SSA, CPD.

1 Introduction

The main object of the paper is multidimensional time series — a set of observed signals $\{x_i(t)\}_{i=1}^m$ where $t \in 1 \dots N$.

Decomposition [1, 2, 3] and forecast [4, 1, 5] methods for sole signal can not be directly transferred to set of signals if their behavior is interconnected. For instance, let's take predator-prey model [6]. The size of one population is dependent on another's and vice-versa. We are further assuming this connection to exist and will give formal definitions in the scope of particular models.

First, RNN model [7, 8] links series with each other and their past values through composition of many nonlinear transformations. This information is encapsulated in hidden vector at each step. With that, one is able to forecast future values.

Second, vector autoregression (VAR) [9, 10] is a linear stochastic model for multidimensional signals. Denoting vector $\mathbf{x}_t = (x_1(t) \dots x_m(t))^\top$ as a series realizations at time t , the further dynamics is following:

$$\mathbf{x}_t = \boldsymbol{\mu} + \sum_{i=1}^p A_i \mathbf{x}_{t-i} + \mathbf{u}_t$$

Here $\boldsymbol{\mu}$ — some constant vector, A_i are matrices $m \times m$, \mathbf{u}_t is a random vector (e.g. white noise $WN(t)$). Signal's connection is defined by A_i transformations, so again each series is dependent on each other and the whole past *linearly*.

Revised solutions allow to make predictions, but have quite a number of hyperparameters to be tuned. They also require exhaustive learning procedures. Furthermore, their structure does not contain explicit way of making series decomposition.

To tackle listed issues, we developed a new method called tSSA which has only two adjustable variables. Computing tensor Canonical polyadic decomposition (CPD) is the only requirement for forecast and decomposition. Next, the approach is an extension of SSA method [11] for multidimensional time series. Due to that, its theory will be based on dynamical systems subtopic — *signal subspace* [12]. The main idea here is to build a phase representation of the observed signals (fig. 1).

In the end, we should mention another SSA modification — mSSA [13]. In contrast, it operates with data only in a way of matrices and matrix decomposition. Nonetheless, both methods share same ideas in forecast and decomposition techniques.

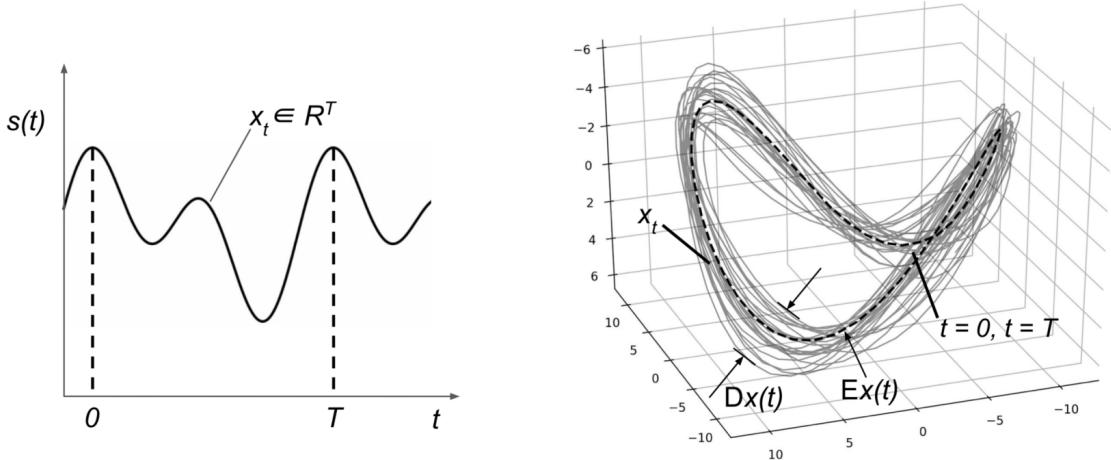


Figure 1: Time series's phase trajectory visualization

The rest of the paper covers the theory and application of our method on the real data. Firstly, the mathematical model will be introduced and basis search in the signal space problem will be stated and resolved. These results will enable us to propose a way of decomposing the signals and make a forecast. Simultaneously, features of obtained solutions will be examined and result in two theorems. Finally, tSSA and mentioned methods will be applied on two datasets: electricity consumption and meteorology observations. We will obtain signal's predictions and additive decompositions. For the latter a special metric will be formulated. All results will be backed with figures and discussion.

2 Problem statement

Let us have observed multidimensional time series $\{x_i(t)\}_{i=1}^m$ generated by some *dynamical system* f . By that we mean a law of coordinates $\mathbf{y} \in X$ evolution in discrete time:

$$\begin{aligned}\mathbf{y}(t+1) &= f(\mathbf{y}(t)), \quad t \in \mathbb{N} \\ \mathbf{y}(0) &= \mathbf{y}_0\end{aligned}$$

Here X is generally a high dimensional smooth manifold. Then, each trajectory of the system spawns our signals through unknown mapping $\varphi : X \rightarrow \mathbb{R}^m$:

$$\varphi(\mathbf{y}(t)) = \mathbf{x}_t \Leftrightarrow \begin{cases} \varphi_1(\mathbf{y}(t)) = x_1(t) \\ \dots \\ \varphi_m(\mathbf{y}(t)) = x_m(t) \end{cases}$$

After that, it is hypothesized that trajectories $\mathbf{y}(t)$ actually lie in a smaller dimension manifold $M \subset X$. So now we should find an embedding of M into \mathbb{R}^L for some L and discover a basis in the image of the embedding. Having done that, we will obtain characterization of the initial system in terms of the standard linear space. The same will be equally applied to all $x_i(t)$.

One signal case

Our future approach roots from SSA method and Takens's theorem [14]. Let's describe it in short. The theorem provides a solution for the stated problem in case of single series: any point $\mathbf{y}(t) \in M$ we associate with the following vector:

$$(\varphi \circ f^{t-L+1}(\mathbf{y}(t)), \dots, \varphi \circ f(\mathbf{y}(t)), \varphi \circ \mathbf{y}(t))^T = (x(t-L+1), \dots, x(t-1), x(t))^T$$

It is called *delay vector* in time t and denoted as $\overleftarrow{\mathbf{x}}_t$. Their dimensionality L should satisfy $L > 2 \cdot \dim(M)$. Function $\varphi(\cdot)$ should also meet some regularity conditions which are assumed to be fulfilled.

So, with signal $x(t)$ of length N we build $N - L + 1$ delay vectors. Corresponding embedding space $\text{Lin}(\{\overleftarrow{\mathbf{x}}_t\})$, so called *signal subspace*, must be low dimensional, meaning $\text{Lin}(\{\overleftarrow{\mathbf{x}}_t\}) \subset \mathbb{R}^L$. Orthonormed basis here is a SVD's U -component of *trajectory matrix* \mathbf{H}_x which is build from delay vectors:

$$\mathbf{H}_x = [\overleftarrow{\mathbf{x}}_1 \dots \overleftarrow{\mathbf{x}}_{N-L+1}]$$

Series interconnection and tSSA method

Теперь обобщим данный подход на несколько временных рядов.

Now let's generalize previous results for the multiple series.

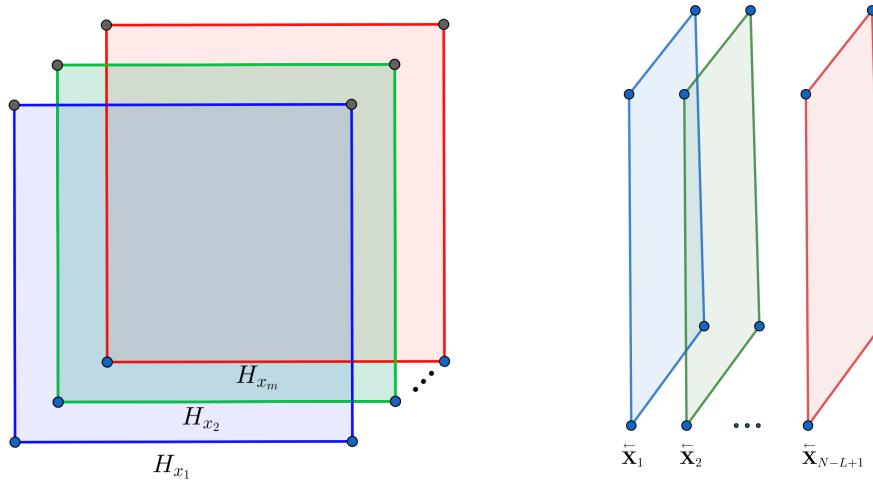


Figure 2: Two views on trajectory tensor. The left is in terms of signals' trajectory matrices $\{x_i(t)\}_{i=1}^m$. The right is in terms of delay matrices.

In this case $\varphi(\cdot)$ is multidimensional. So instead of the single delay vector we should consider their set for all m signals in time t . That means the embedding image now contains so called *delay matrices* $(\overleftarrow{\mathbf{x}}_{1t} \dots \overleftarrow{\mathbf{x}}_{mt}) := \overleftarrow{\mathbf{X}}_t$. Next, instead of trajectory matrix we should introduce *trajectory tensor* \mathbf{T} . It is constructed by putting delay matrices along the second dimension

of the tensor, fig. 2 (left). This procedure is equivalent to the one for \mathbf{H}_x . But our tensor has an important property: \mathbf{T} can also be built as an alignment of signals' trajectory matrices \mathbf{H}_{x_i} along the third dimension, fig. 2 (right).

Then, reasoning analogues to the one signal case we obtain *signals' set subspace* as a linear span of delay matrices. But we were supposed to built the subspace for each signal individually.

It's time to give a definition for signals' interrelation in our model's scope.

Definition. We call a set of time series *interconnected* if they all share the common subspace with the same basis.

Now we are ready to find this shared elements for signals. Note that to do so we just need to decompose each signal's trajectory matrix through the same set of factors. Finally, let's apply CPD to our trajectory tensor and have a look at its slices by third dimension.

$$\mathbf{T} = \sum_{i=1}^r \mathbf{a}_i \otimes \mathbf{b}_i \otimes \mathbf{c}_i \Leftrightarrow \begin{cases} \mathbf{H}_{x_1} = \sum_{i=1}^r \boldsymbol{\sigma}_{x_1}(i) \cdot \mathbf{a}_i \mathbf{b}_i^\top \\ \mathbf{H}_{x_2} = \sum_{i=1}^r \boldsymbol{\sigma}_{x_2}(i) \cdot \mathbf{a}_i \mathbf{b}_i^\top \\ \dots \\ \mathbf{H}_{x_m} = \sum_{i=1}^r \boldsymbol{\sigma}_{x_m}(i) \cdot \mathbf{a}_i \mathbf{b}_i^\top \end{cases} \quad (1)$$

What have we got? CPD is defined by tensor rank r and a set of vectors which we can pack in matrices: $A = [\mathbf{a}_1 \dots \mathbf{a}_r]; B = [\mathbf{b}_1 \dots \mathbf{b}_r]; C = [\mathbf{c}_1 \dots \mathbf{c}_r]$. Row k of matrix C is denoted as $\boldsymbol{\sigma}_{x_k}$. It has the sense of singular values but can be negative.

After, from CPD definition as a (1) kind decomposition with minimal r we can derive that A, B, C are full-rank matrices. So, signals' shared subspace is $\text{Lin}(\{\mathbf{a}_i\})$ with unified basis $\{\mathbf{a}_i\}_{i=1}^r$ which is not orthonormed in general. Then, back in the problem statement we assumed dimensionality of the embedding to be rather small. Therefore relation $r \ll L$ should be true.

Lastly, let's note that *rows* of \mathbf{H}_{x_k} are also delay vectors but have another dimension of $N - L + 1$. So the shared subspace for them would be $\text{Lin}(\{\mathbf{b}_i\})$. This result proves our approach to be consistent with choice of $\overleftarrow{\mathbf{x}}_{k,t}$. As a remark, mSSA lacks such feature.

What to the signals' link, it is regulated by $\boldsymbol{\sigma}_{x_k}$. For instance, if all the rows have zero elements in non-overlapping positions then our built subspace breaks down into a direct sum of individual signal's subspaces. It is total link absence case. The exact opposite situation is when $\boldsymbol{\sigma}_{x_k}$ are non-zero for all series.

Time series decomposition

We can break series into several components using the conception: *decomposition of trajectory matrix \mathbf{H}_{x_k} defines decomposition of related signal*. Its first part implies representation of \mathbf{H}_{x_k} as a sum of factors (1). This technique comes from SSA. Because all matrices are similarly decomposed, we can focus on the task for a single series.

Besides, we should point out the main feature of trajectory matrices — *hankelness*. That simply means equal elements along each anti-diagonal. Every series of length N bijectionally corresponds to hankel matrix of size $L \times (N - L + 1)$.

Now, decomposition procedure is following: we assume factors of \mathbf{H}_{x_k} to be able to pack in s groups in a special way. That means if we sum all factors within each group the result matrices C_1, \dots, C_s will be hankel. This situation would perfectly correspond to representation of $x_k(t)$ as a sum of s signals. Unfortunately, it is almost infeasible even with trivial series [11], so every C_i should be additionally *hankelized*. This operation average each matrix's anti-diagonal so it becomes hankel. Let's denote the operator as $\text{Hankel}(\cdot)$.

The whole algorithm can be written as a chain of identical expressions:

$$\begin{aligned}
\mathbf{H}_{x_k} &\stackrel{1}{=} \sum_{i=1}^r \boldsymbol{\sigma}_{x_k}(i) \cdot \mathbf{a}_i \mathbf{b}_i^\top \stackrel{2}{=} \sum_{i \in \mathbb{I}_1} \boldsymbol{\sigma}_{x_k}(i) \cdot \mathbf{a}_i \mathbf{b}_i^\top + \dots + \sum_{i \in \mathbb{I}_s} \boldsymbol{\sigma}_{x_k}(i) \cdot \mathbf{a}_i \mathbf{b}_i^\top \stackrel{3}{=} \\
&\stackrel{3}{=} C_1 + \dots + C_s \stackrel{4}{=} \text{Hankel}(C_1) + \dots + \text{Hankel}(C_s) \Leftrightarrow x_k(t) = c_1(t) + \dots + c_s(t) \quad (2)
\end{aligned}$$

Here $\mathbb{I}_1 \sqcup \dots \sqcup \mathbb{I}_s = \{1, \dots, r\}$ — chosen groups of indices. Actually, the forth equality needs some extra validation. To make it, first consider hankel operator's property:

Lemma 1. *Hankel(\cdot) operator is linear.*

Proof. Having matrices A and B with the same size, let's look at their elements from one anti-diagonal a_1, \dots, a_n and b_1, \dots, b_n . Then this anti-diagonal in $\text{Hankel}(A + B)$ is written as $\frac{1}{n} \sum_{i=1}^n (a_i + b_i) = \frac{1}{n} \sum_{i=1}^n a_i + \frac{1}{n} \sum_{i=1}^n b_i$, resulting in a sum of $\text{Hankel}(A)$ and $\text{Hankel}(B)$ anti-diagonals.

Secondly, having a scalar α it is also trivial that $\frac{1}{n} \sum_{i=1}^n \alpha \cdot a_i = \alpha \frac{1}{n} \sum_{i=1}^n a_i$. Then we finally derive $\text{Hankel}(\alpha A) = \alpha \text{Hankel}(A)$. \square

Returning to the forth equality in (2), we can combine this lemma and the fact that \mathbf{H}_{x_k} is itself hankel to get a complete justification. This outputs in the correctness of series decomposition procedure for any chosen grouping of factors.

Optimal decomposition and its computational complexity

Even though decomposition algorithm from the previous subsection is correct for any factors' grouping we might want to find the best one. This would mean every C_i to be as close to hankel matrix as possible. Having that the formulated decomposition concept would be fully accomplished and signal's components could be naturally extracted. Let's rewrite it as some optimization task.

Consider the first equality from (2) and itself with hankel operator applied. Also recall that \mathbf{H}_{x_k} is hankel. We obtain:

$$\begin{cases} \mathbf{H}_{x_k} = \sum_{i=1}^r \boldsymbol{\sigma}_{x_k}(i) \cdot \mathbf{a}_i \mathbf{b}_i^\top \\ \mathbf{H}_{x_k} = \sum_{i=1}^r \text{Hankel}(\boldsymbol{\sigma}_{x_k}(i) \cdot \mathbf{a}_i \mathbf{b}_i^\top) \end{cases}$$

Subtract the second equation from the first and denote $H_i = \boldsymbol{\sigma}_{x_k}(i) \cdot \mathbf{a}_i \mathbf{b}_i^\top - \text{Hankel}(\boldsymbol{\sigma}_{x_k}(i) \cdot \mathbf{a}_i \mathbf{b}_i^\top)$. This can be sensed as residual between initial matrix and its hankelized version. So we have:

$$H_1 + \dots + H_r = 0 \Leftrightarrow H_r = - \sum_{j=1}^{r-1} H_j \quad (3)$$

What we have done is reformulated condition on C_i to be hankel in terms of residual matrices summing into zero inside groups: $\sum_{k \in \mathbb{I}_i} H_k = 0, \forall i \in 1, \dots, s$.

Now let's simplify our case by decreasing the number of desired components from s to two. If we worked out the easier problem we could consecutively dissect components into two more and have as much of them as we want.

Next, (3) shows all residual matrices already total in zero. Therefore we can discard, for example, H_r as group for it would be defined automatically if having groups for the rest factors. Let's introduce indicator-variables $\beta_j \in \{0, 1\}$ showing in what of two groups factor j lies.

Finally, we assume all H_i to be non-zero; otherwise we already have needed dissection. This last condition implies each group to have at least two residual matrices. So we are ready for the optimal decomposition problem statement:

$$\begin{cases} \sum_{j=1}^{r-1} \beta_j H_j = 0 \\ \beta_j \in \{0, 1\}, \forall j \in 1, \dots, r \\ \sum_{i=1}^{r-1} \beta_j \geq 2 \end{cases} \quad (4)$$

To rewrite it in a more familiar way let's vectorize each H_i and put resulting vectors in a matrix Λ . Furthermore, introduce $\boldsymbol{\beta} = (\beta_1 \dots \beta_{r-1})$. Then, recall from previous subsection that it is highly unlikely for residual matrices to total in zero within all groups. The most we can is to make them as close to zero as possible. So restated problem:

$$\begin{cases} \|\Lambda \boldsymbol{\beta}\| \rightarrow \min_{\boldsymbol{\beta}} \\ \beta_j \in \{0, 1\}, \forall j \in 1, \dots, r \\ \sum_{i=1}^{r-1} \beta_j \geq 2 \end{cases} \quad (5)$$

Let's declare solution of this task *optimal series decomposition*. It is represented as Integer Least Squares problem which is proved to be NP-hard [15]. In short, subsection's output is:

Theorem 1. *Optimal series decomposition as special grouping of trajectory matrix's factors search is NP-hard.*

In spite of the result, methods and heuristics exist to reduce ILS to computationally effective objectives [16].

Forecasting

In our model prediction making comes down to restoration of delay vectors for every signal in a set. Because we have the shared basis among series $\{\mathbf{a}_i\}_{i=1}^r$ we can process them separately.

If $\{\mathbf{a}_i\}_{i=1}^r$ was orthonormal we could inherit SSA prediction technique [11]. But we don't have this warranty so we are going to adopt it for our case.

Having input series $x(t)$ we want to forecast it for one step further. Delay vector $\overleftarrow{\mathbf{x}}_{N+1} = (x(N-L) \dots x(N+1))^T$, with the last component unknown, belongs to the $\text{Lin}(\{\mathbf{a}_i\})$. In terms of full-ranked matrix $A \in \mathbb{R}^{L \times r}$ (see [Series interconnection and tSSA method](#)) it can be written as:

$$\begin{aligned} \overleftarrow{\mathbf{x}}_{N+1} = A\boldsymbol{\lambda} &\Leftrightarrow \begin{cases} \mathbf{x}_{kn} = A_{kn}\boldsymbol{\lambda} \\ x(N+1) = \mathbf{a}_{pr}^T \boldsymbol{\lambda} \end{cases}, \text{ where} \\ A &= \begin{pmatrix} A_{kn} \\ \mathbf{a}_{pr}^T \end{pmatrix} \\ \overleftarrow{\mathbf{x}}_{N+1} &= (\mathbf{x}_{kn} \ x(N+1))^T \end{aligned} \quad (6)$$

Here variable $\boldsymbol{\lambda} \in \mathbb{R}^r$. Delay vector and matrix are broken into two blocks associated with the *known* and *unknown* part of the series. To make prediction we need to find $\boldsymbol{\lambda}$ so we focus on the first obtained equation.

The matrix of this linear system $A_{kn} \in \mathbb{R}^{(L-1) \times r}$ has a rank at least $r - 1$ by construction. But we assume elimination of one row in A will actually keep the rank full. As experiments

show this is not burdensome. After that, we note that the linear system itself is overdetermined due to the model hypothesis $r \ll L$. Therefore it is reasonable to get solution in a least squares sense: $\boldsymbol{\lambda} = (A_{kn}^T A_{kn})^{-1} A_{kn}^T \mathbf{x}_{kn}$. In the end, forecast is given by the formula:

$$x(N+1) = \mathbf{a}_{pr}^T (A_{kn}^T A_{kn})^{-1} A_{kn}^T \mathbf{x}_{kn} \quad (7)$$

Here $\mathbf{a}_{pr}^T (A_{kn}^T A_{kn})^{-1} A_{kn}^T = \mathbf{d}^T$ — row vector which once computed can be reused to make prediction for arbitrary horizon ahead.

Now let's rewrite (7) in terms of $x(t)$:

$$x(t) = \sum_{i=1}^{L-1} d_i \cdot x(t-i)$$

This expression leads to the final result about the forecast behavior:

Theorem 2. *The model of tSSA's forecast is autoregressive AR($L-1$). It is fully characterized by coefficients d_i found in (7).*

3 Computational experiment

Goals and setting

In the current section we will demonstrate application of our method for decomposition and forecast of multidimensional time series. We will also compare it with other models: mSSA, VAR, RNN. The latter two are only able for prediction. The mSSA has a lot in common with our approach and is valid for both tasks.

To measure quality of the decomposition we need special metrics:

Definition. *Absolute hankel error* of matrix M is

$$\text{AHE} = \|M - \text{Hankel}(M)\|_2$$

Definition. *Relative hankel error* of matrix M is

$$\text{RHE} = \frac{\text{AHE}}{\|M\|_2}$$

AHE can be imagined as a total standard deviation of matrix's anti-diagonals. RHE is its more convenient modification which will be used further.

Now, recall that series decomposition on additive components is equivalent to trajectory matrix decomposition, viz. obtaining of matrices C_i (see [Time series decomposition](#)). The RHE will be calculated for them as well as mean RHE for every series in a set — $\overline{\text{RHE}}_{ts}$ and the mean among all series — $\overline{\text{RHE}}$.

To measure forecast quality we will be using standard metrics: mean squared error MSE_{ts} and mean absolute percentage error MAPE_{ts} . Their averaging for all series are denoted as $\overline{\text{MSE}}$ and $\overline{\text{MAPE}}$.

Two sets of data will be involved. First, electricity consumption paired with its generation price (fig. 3). Second, three series describing weather conditions in Berlin from 1980 to 1990: temperature, precipitation and air pressure (fig. 4). We assume signals' inter-dependency based on economical/physical reasons as it is important premise for our approach (see [Series interconnection and tSSA method](#)). Then, individual series has a length $\approx 3 \cdot 10^3$ where last 20% are deferred test samples for prediction evaluation.

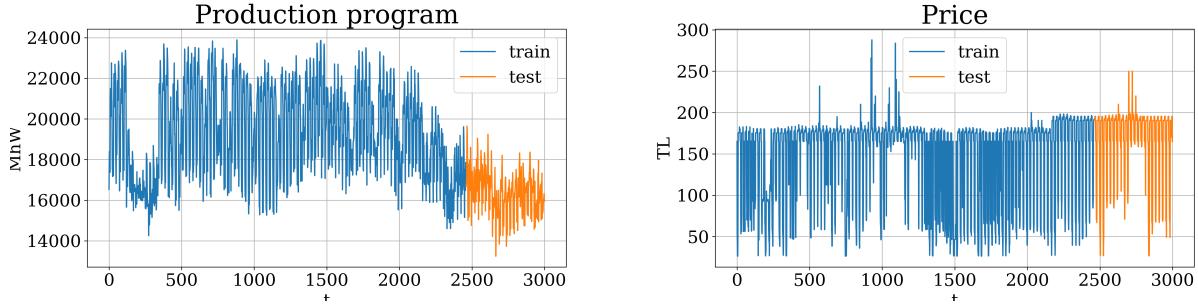


Figure 3: Time series for electricity consumption and its price

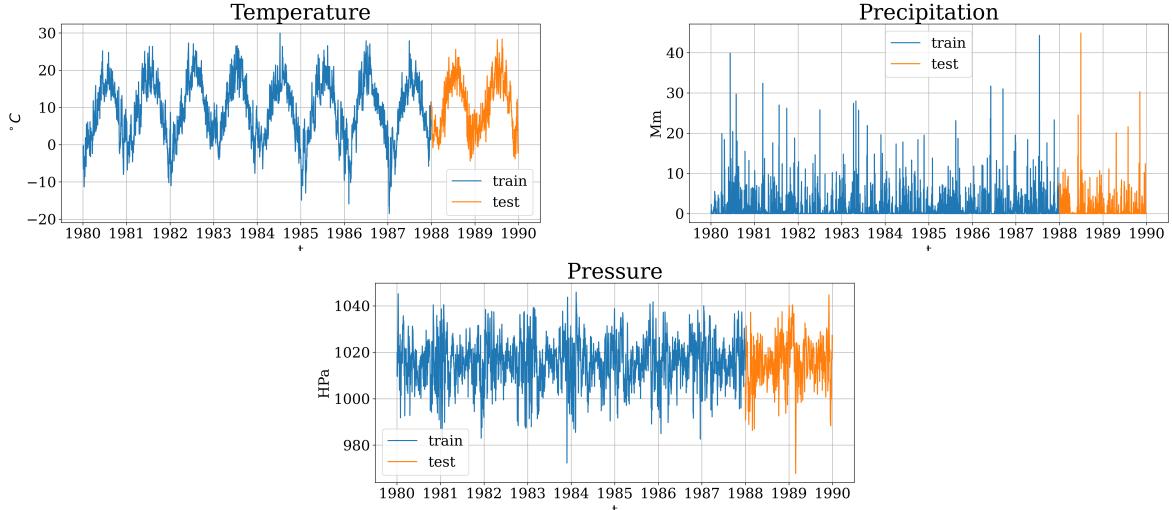


Figure 4: Time series for temperature, precipitation and air pressure from meteostation in Berlin.

Let's describe peculiarities of tSSA application. Firstly, tensor rank computation is NP-hard [17], so its number will be explicitly indicated whenever needed. Secondly, the used algorithm for CPD is ALS [18] and it has an approximation error. Thanks for that trajectory matrices' decomposition (1) is not exact. This relative errors will be specified below. Thirdly, ILS problem (5) must be solved to find series decomposition. But matrix involved in it has a large row dimension. To boost computation we will cut it to several hundreds which is still much bigger then the size of sought parameter's vector. The used solver is SCIP [19].

Finally, let's fix a value L unified for all models. It is generally responsible for the size of series' history on which methods can be conditioned. For tSSA and mSSA it was described in the beginning of the section 2. For RNN it is the length of encoder. For VAR it is the maximum lag number. We set $L = 500$.

Results and discussion

We begin with forecast results. First, relation between prediction metrics for tSSA and trajectory tensor's rank is depicted on the fig. 5 and 6. The minimum point and overall pattern are similar for MSE and MAPE. Furthermore, overtraining effect can be seen with especially sharp surge on the weather data. Nonetheless, CPD approximation error decease monotonously, fig. 7.

The best tSSA forecast is shown on fig. 8 and 9. For electricity CPD rank is rather high and so the order of autoregression forecast model (see Forecasting). It allows for the complex prediction. On the other hand, for the weather high rank led to unstable and unbound forecast, but with the small rank one has few degrees of freedom. Consequently, we can only obtain some

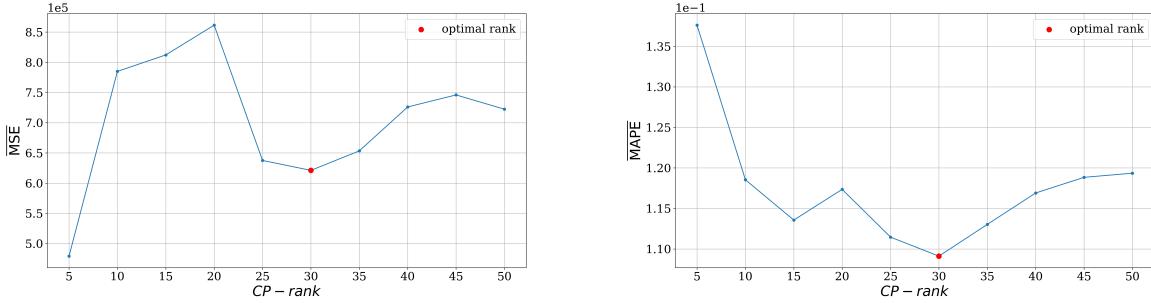


Figure 5: $\overline{\text{MSE}}$ and $\overline{\text{MAPE}}$ metrics for tSSA forecast depending on CPD rank. Optimal point is marked with red. Electricity data.

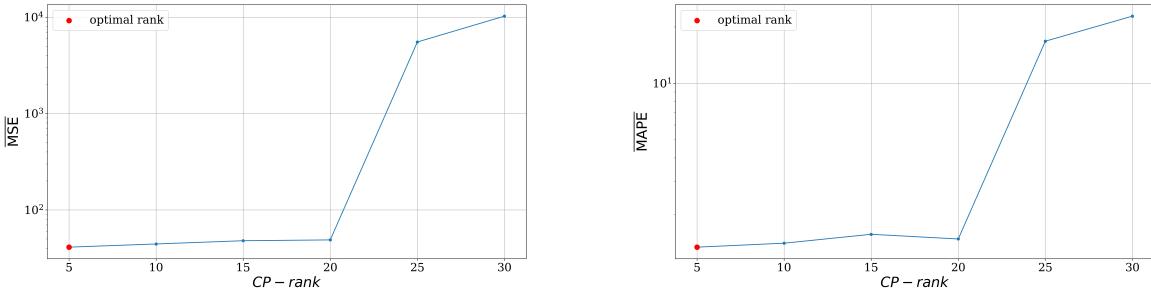


Figure 6: $\overline{\text{MSE}}$ and $\overline{\text{MAPE}}$ metrics for tSSA forecast depending on CPD rank. Optimal point is marked with red. Weather data.

mean values for the future.

The final forecast metrics for all models are in tab. 1 and 2. Our method gave the best output. Close performance was shown by mSSA. The VAR model appeared unstable for the chosen prediction horizon and RNN could only learn the constant function from data.

Table 1: Prediction quality of models for electricity data

	<i>tSSA</i>	<i>mSSA</i>	<i>VAR</i>	<i>RNN</i>
$\overline{\text{MSE}}_{\text{Production}}, 10^6$	1.24	1.51	7.81	2.70
$\overline{\text{MSE}}_{\text{Price}}, 10^3$	0.88	1.03	4.85	30.0
$\overline{\text{MSE}}, 10^6$	0.62	0.75	3.91	135.00
$\overline{\text{MAPE}}_{\text{Production}}$	0.054	0.060	0.137	0.999
$\overline{\text{MAPE}}_{\text{Price}}$	0.164	0.170	0.360	1.004
$\overline{\text{MAPE}}$	0.109	0.115	0.249	1.002

Now we move to the decomposition. Fig. 10 illustrates relation between quality metrics and CPD rank. In both cases $\overline{\text{RHE}}$ rapidly reaches a minimum and then does not change or even increase. Recalling the same result for the forecast we can conclude that tSSA has the greatest generalization capacity for the small tensor ranks. At the same time, the rank is proportional to the signal subspace dimensionality (see section 2). So this approves the major premise of our method — small real dimension of the data.s

One the fig. 11, 12 one can see best tSSA's decomposition on two components. Due to rather long computation of ILS we do not make more. And it is not necessary to catch the notion of shared basis for observed signals in a set. For example, decomposition for electricity generation and its price are almost identical up to the bias and scale. The same situation is for temperature and precipitation from the weather data, but air pressure got its own components. Now, let's

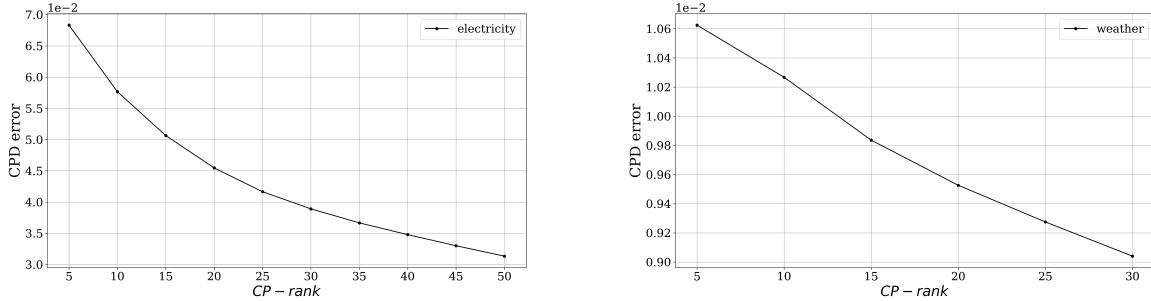


Figure 7: Relative CPD approximation errors depending on CPD rank. Left is for electricity, right is for weather.

Table 2: Prediction quality of models for weather data

	$tSSA$	$mSSA$	VAR	RNN
\overline{MSE}_{Temp}	14.16	20.12	40.67	-
\overline{MSE}_{Prec}	11.02	11.23	22.84	-
\overline{MSE}_{Pres}	98.02	101.23	156.75	-
\overline{MSE}	41.067	44.193	73.420	-
\overline{MAPE}_{Temp}	0.850	0.626	0.468	-
\overline{MAPE}_{Prec}	3.151	2.834	4.637	-
\overline{MAPE}_{Pres}	0.007	1.254	3.983	-
\overline{MAPE}	1.336	1.571	3.029	-

interpret obtained results. On the fig. 12 method extracted two periodical parts for each signal with large and much smaller amplitudes. Some noise is present in each component. On the fig. 11 we have two high-frequency harmonics with different amplitude and bias.

In comparison, we present mSSA’s decomposition on five components, fig. 13 and 14. There is no need to solve heavy ILS problem or pick over CPD rank, but it is a manual procedure based on singular values analysis of some matrix [11]. Nonetheless, the gained components are simpler and more diverse — algorithm extracted several trends and harmonics. Moreover, mSSA has slight advantage in decomposition quality as described in tab. 3, 4. To conclude, the complexity of tSSA decomposition problem is the major stumbling block of our method which requires development of its own solutions and heuristics. For the record, mSSA has already had those [11].

Table 3: Decomposition quality of models for electricity data

	$tSSA$	$mSSA$
$\overline{RHE}_{Production}$	0.507	0.308
\overline{RHE}_{Price}	0.511	0.31
\overline{RHE}	0.509	0.309

4 Conclusion

Tensor method tSSA is devoted to prediction and decomposition of multidimensional time series with substantial interconnection between them. It is easy to use — there are only two adjustable parameters which does not require heavy learning procedures. Besides, the theory behind is based on the general framework of dynamical systems and demands only low-dimensional

Table 4: Decomposition quality of models for weather data

	tSSA	mSSA
$\overline{\text{RHE}}_{\text{Temp}}$	0.512	0.467
$\overline{\text{RHE}}_{\text{Prec}}$	0.508	0.538
$\overline{\text{RHE}}_{\text{Pres}}$	0.542	0.502
$\overline{\text{RHE}}$	0.521	0.502

representation of data. When applied to real sets of signals, these assumptions were confirmed as well as the neural and statistical models were outperformed by our approach. Nonetheless, a bottleneck was discovered which is NP-hardness of optimal series decomposition. Resolving this problem or finding another way to build signal's components are the major ways for further improvement.

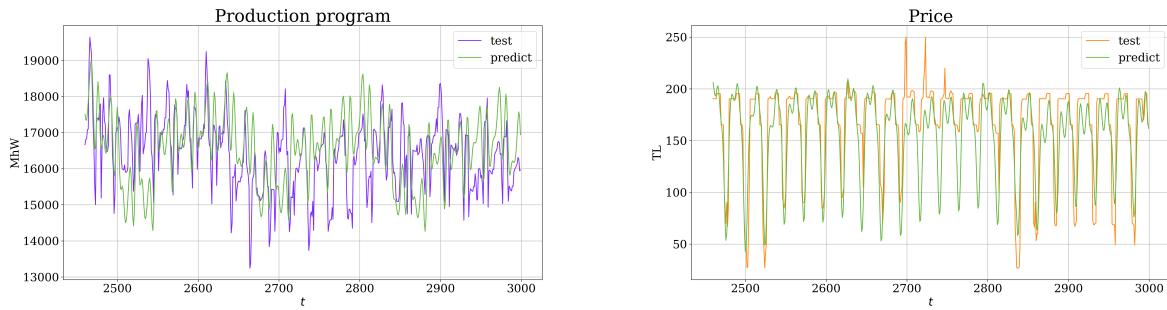


Figure 8: tSSA forecast for electricity data. CPD rank = 30

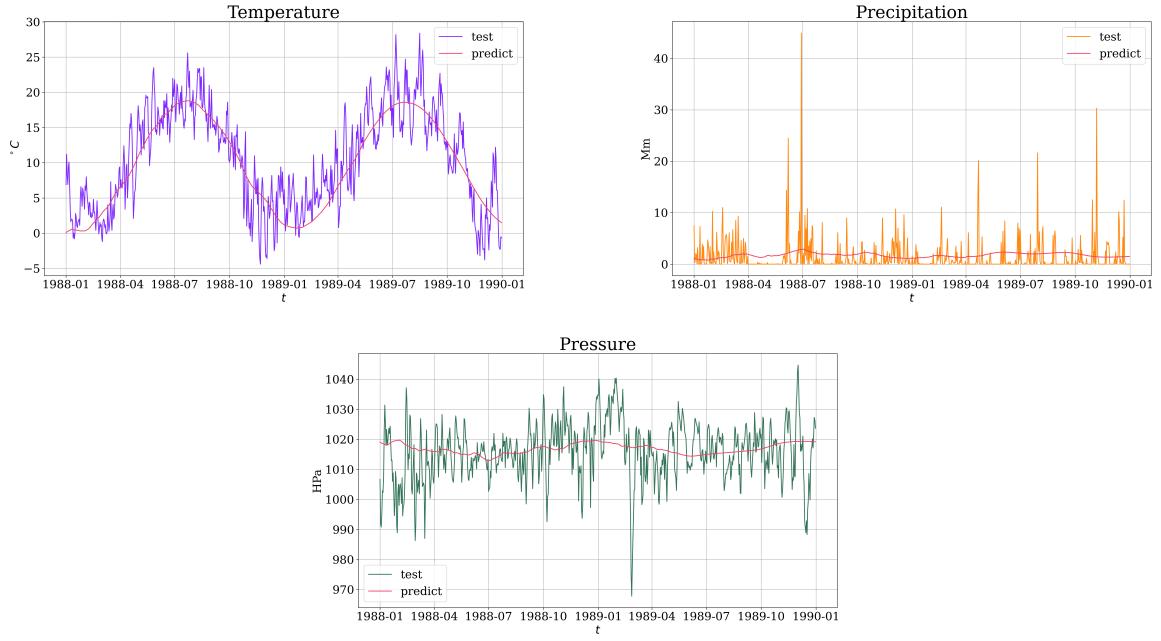


Figure 9: tSSA forecast for weather data. CPD rank = 5

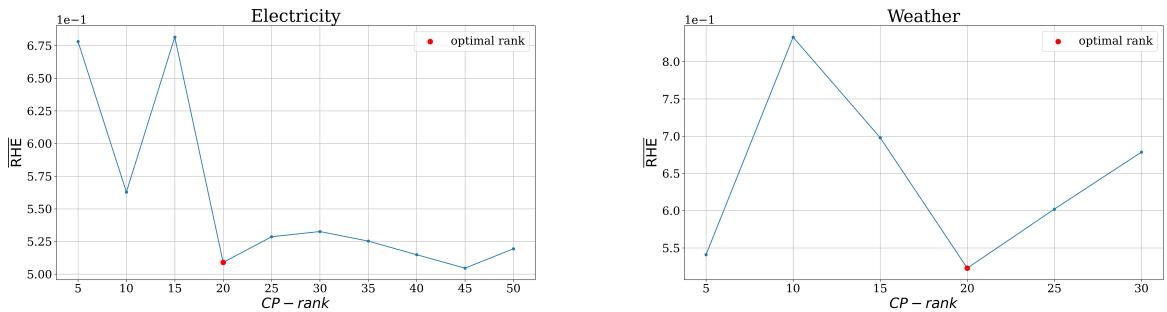


Figure 10: Relation between $\overline{\text{RHE}}$ metrics and CPD rank. The left is for electricity data, the right is for weather data. Optimal point is marked with red

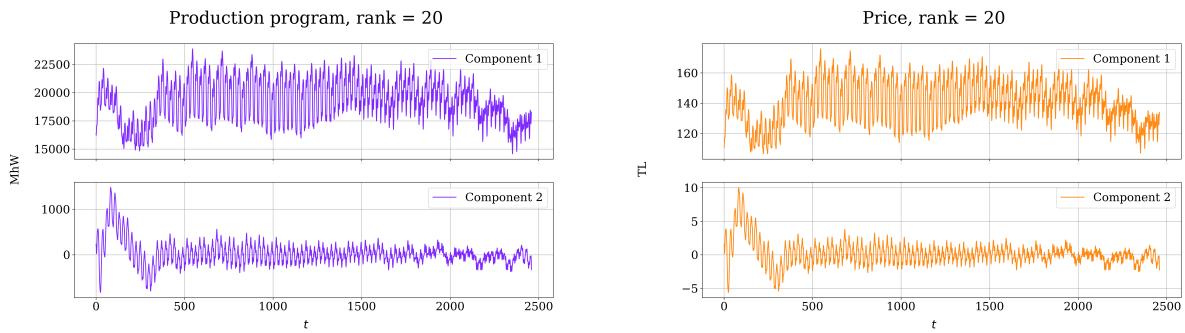


Figure 11: tSSA decomposition for electricity data. CPD rank = 20

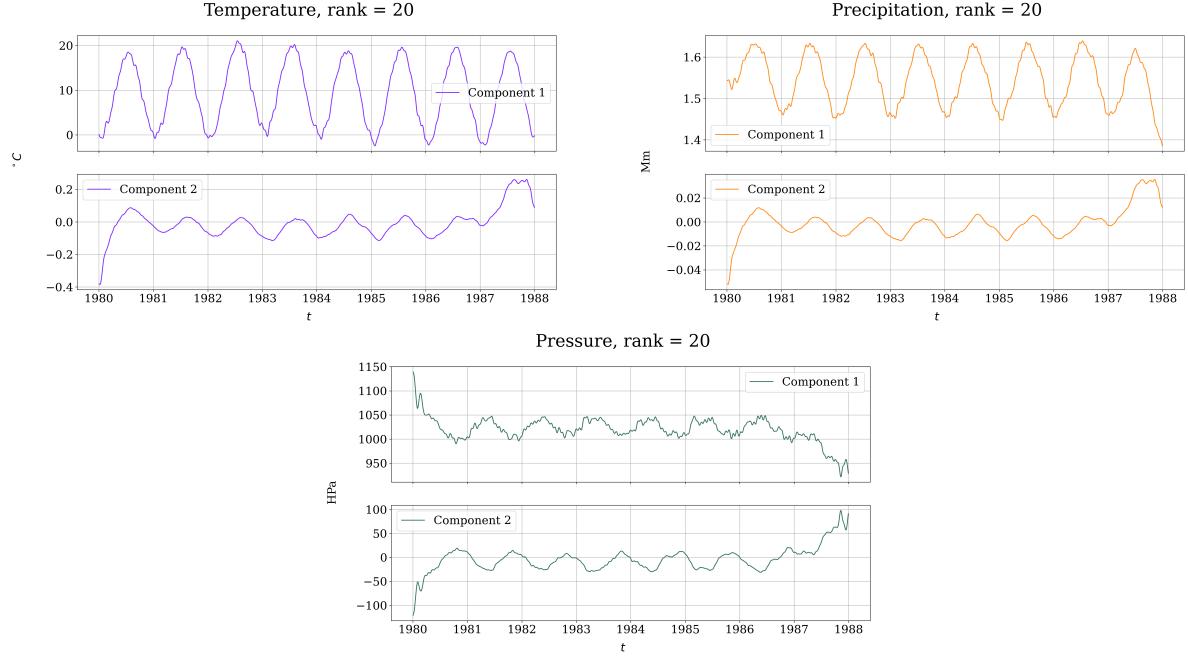


Figure 12: tSSA decomposition for weather data. CPD rank = 20

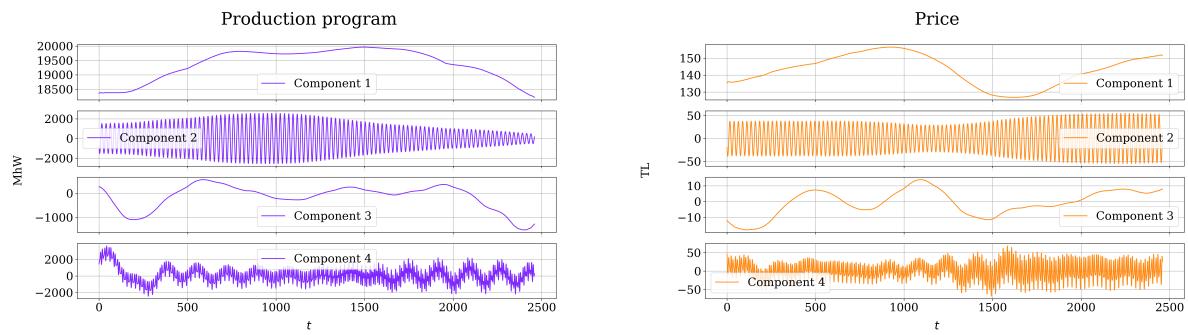


Figure 13: mSSA decomposition for electricity data

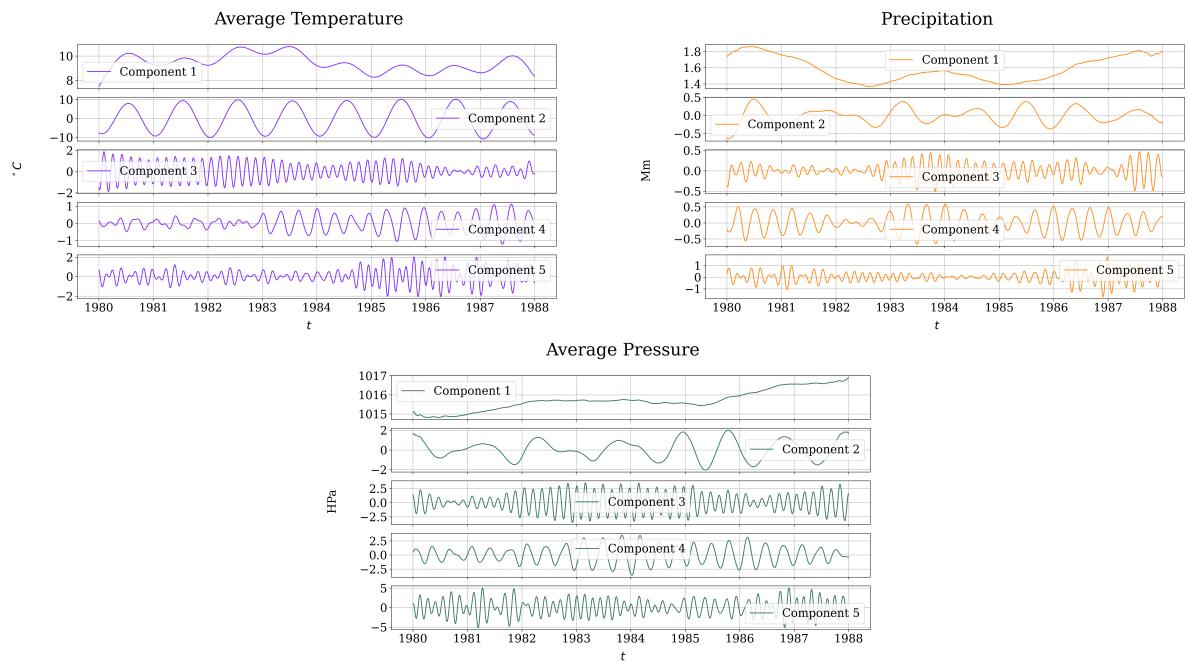


Figure 14: mSSA decomposition for weather data

References

- [1] W. Enders and John Wiley & Sons. *Applied Econometric Time Series*. Wiley series in probability and statistics. Wiley, 2010. ISBN: 9788126543915. URL: <https://books.google.ru/books?id=vzJ0CgAAQBAJ>.
- [2] Estela Dagum and Silvia Bianconcini. *Seasonal Adjustment Methods and Real Time Trend-Cycle Estimation*. Aug. 2016. ISBN: 978-3-319-31820-2.
- [3] Robert B. Cleveland et al. “STL: A Seasonal-Trend Decomposition Procedure Based on Loess (with Discussion)”. In: *Journal of Official Statistics* 6 (1990), pp. 3–73.
- [4] Robin John Hyndman and George Athanasopoulos. *Forecasting: Principles and Practice*. English. 2nd. Australia: OTexts, 2018.
- [5] Granville Tunnicliffe Wilson. “Time Series Analysis: Forecasting and Control,5th Edition, by George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel and Greta M. Ljung, 2015. Published by John Wiley and Sons Inc., Hoboken, New Jersey, pp. 712. ISBN: 978-1-118-67502-1”. In: *Journal of Time Series Analysis* 37 (Mar. 2016), n/a–n/a. DOI: [10.1111/jtsa.12194](https://doi.org/10.1111/jtsa.12194).
- [6] Vito Volterra and Marcel Brelot. “Lecons sur la théorie mathématique de la lutte pour la vie”. In: 1931. URL: <https://api.semanticscholar.org/CorpusID:153535842>.
- [7] Sepp Hochreiter and Jürgen Schmidhuber. “Long Short-Term Memory”. In: *Neural Computation* 9.8 (Nov. 1997), pp. 1735–1780. ISSN: 0899-7667. DOI: [10.1162/neco.1997.9.8.1735](https://doi.org/10.1162/neco.1997.9.8.1735). eprint: <https://direct.mit.edu/neco/article-pdf/9/8/1735/813796/neco.1997.9.8.1735.pdf>. URL: <https://doi.org/10.1162/neco.1997.9.8.1735>.
- [8] Ahmed Tealab. “Time series forecasting using artificial neural networks methodologies: A systematic review”. In: *Future Computing and Informatics Journal* 3.2 (2018), pp. 334–340. ISSN: 2314-7288. DOI: <https://doi.org/10.1016/j.fcij.2018.10.003>. URL: <https://www.sciencedirect.com/science/article/pii/S2314728817300715>.
- [9] Dimitrios Asteriou and Stephen Hall. “Vector Autoregressive (VAR) Models and Causality Tests”. In: Jan. 2016, pp. 333–346. ISBN: 978-1-137-41546-2. DOI: [10.1057/978-1-137-41547-9_15](https://doi.org/10.1057/978-1-137-41547-9_15).
- [10] Arnold Zellner. “An Efficient Method of Estimating Seemingly Unrelated Regressions and Tests for Aggregation Bias”. In: *Journal of the American Statistical Association* 57.298 (1962), pp. 348–368. DOI: [10.1080/01621459.1962.10480664](https://doi.org/10.1080/01621459.1962.10480664). eprint: <https://www.tandfonline.com/doi/pdf/10.1080/01621459.1962.10480664>. URL: <https://www.tandfonline.com/doi/abs/10.1080/01621459.1962.10480664>.
- [11] D. Stepanov and N. Golyandina. “SSA-based approaches to analysis and forecast of multi-dimensional time series”. In: *Proceedings of the 5th St.Petersburg Workshop on Simulation*. 2005, pp. 293–298.
- [12] Hamid KRIM. “Two Decades of Array Signal Processing Research”. In: *IEEE Signal Processing Magazine* 0 (1994), pp. 67–94. URL: <https://cir.nii.ac.jp/crid/1572261550523548160>.
- [13] Hossein Hassani and Rahim Mahmoudvand. “Multivariate singular spectrum analysis: A general view and new vector forecasting approach”. In: *International Journal of Energy and Statistics* 01 (Apr. 2013). DOI: [10.1142/S2335680413500051](https://doi.org/10.1142/S2335680413500051).
- [14] Floris Takens. “Detecting Strange Attractors in Turbulence”. In: *Dynamical Systems and Turbulence, Warwick 1980*. Ed. by David Rand and Lai-Sang Young. Vol. 898. Lecture Notes in Mathematics. Berlin: Springer, 1981. Chap. 21, pp. 366–381. ISBN: 978-3-540-11171-9. DOI: [10.1007/bfb0091924](https://doi.org/10.1007/bfb0091924). URL: <http://dx.doi.org/10.1007/bfb0091924>.

- [15] P. van Emde-Boas. *Another NP-complete partition problem and the complexity of computing short vectors in a lattice*. Report. Department of Mathematics. University of Amsterdam. Department, Univ., 1981.
- [16] Erik Grafarend, Silvelyn Zwanzig, and Joseph Awange. “Integer Least Squares”. In: *Applications of Linear and Nonlinear Models: Fixed Effects, Random Effects, and Total Least Squares*. Cham: Springer International Publishing, 2022, pp. 579–603. ISBN: 978-3-030-94598-5. DOI: [10.1007/978-3-030-94598-5_16](https://doi.org/10.1007/978-3-030-94598-5_16). URL: https://doi.org/10.1007/978-3-030-94598-5_16.
- [17] Johan Håstad. “Tensor rank is NP-complete”. In: *Journal of Algorithms* 11.4 (1990), pp. 644–654. ISSN: 0196-6774. DOI: [https://doi.org/10.1016/0196-6774\(90\)90014-6](https://doi.org/10.1016/0196-6774(90)90014-6). URL: <https://www.sciencedirect.com/science/article/pii/0196677490900146>.
- [18] Tamara Kolda and Brett Bader. “Tensor Decompositions and Applications”. In: *SIAM Review* 51 (Aug. 2009), pp. 455–500. DOI: [10.1137/07070111X](https://doi.org/10.1137/07070111X).
- [19] Suresh Bolusani et al. *The SCIP Optimization Suite 9.0*. ZIB-Report 24-02-29. Zuse Institute Berlin, Feb. 2024. URL: <https://nbn-resolving.org/urn:nbn:de:0297-zib-95528>.