

Tensor decomposition and forecasting for multivariate time series

Kirill Semkin^{1*} and Vadim Strijov^{2*}

¹*Forecsys® Moscow Russia.

²*Forecsys® Moscow Russia.

*Corresponding author(s). E-mail(s): kirill.semkin32@mail.ru;
strijov@ccas.ru;

Keywords: time series, decomposition, forecast, SSA, CPD

Processing of multidimensional time series is associated with an additional task of determining dependencies between signals. Its inclusion in models boosts the quality of forecasts. On the other hand, taking this dependency into account makes models more complex and less interpretable. The paper proposes a non-parametric method based on tensor data representation and Singular Spectrum Analysis (SSA). It derives a time series decomposition technique and an explicit forecast model. Finally, it applies an elaborated theory to electricity consumption and inertial measurement unit datasets. It compares the obtained quality of forecast with mSSA, VAR, and RNN models.

1 Introduction

The main object of the paper is multivariate time series. It is a set of m series $\{x_i(t)\}_{i=1}^m$ observed in discrete time $t \in 1, \dots, N$. We state two problems concerning them. First, to make a *forecast* is to estimate future values $\{x_i(T)\}_{i=1}^m$ of the series at time $T > N$. Second, to build an additive *decomposition* is to represent each signal in a set as a sum of several components: $x_i(t) = c_1(t) + \dots + c_s(t)$, $\forall i \in 1, \dots, m$.

For the decomposition in a single-variate time series case the papers [1–3] introduce seasonal-trend-cycle techniques. For the forecast the authors in [5] propose autoregressive methods as well as works [1, 4] suggest exponential smoothing, regression models, and neural networks. However, these methods can not be transferred to the multivariate series straightforwardly if they are interdependent. The paper [7] models the size of prey-and-predator populations as a coupled system of differential equations. The dynamics of one population depends directly on the other's and vice-versa. Therefore the forecast can not be made separately.

Some methods both take into account multivariate time series interdependence and make the forecast. First, *recurrent neural network* (RNN) [8] connects series and their past values through several layers of nonlinear transformations. This information is encapsulated in the hidden state vector at each time step. With that, one is able to forecast future values [9].

Second, the *vector autoregression* (VAR) [10, 11] is a linear stochastic model for multivariate series. Denote vector $\mathbf{x}_t = (x_1(t) \dots x_m(t))^\top$ as a series realizations at time t , their further dynamics

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

Here $\boldsymbol{\mu}$ is a constant vector, A_i are $m \times m$ matrices, \mathbf{u}_t is a random vector (e.g. white noise $WN(t)$). The time series interdependence is defined by A_i matrices, so each series depends on each other and their past linearly.

The RNN and VAR methods are able to make the forecast, but, firstly, have a large number of learnable and structural parameters. That means their underlying models have an extensive complexity and their architecture needs to be picked over and tuned. Secondly, the methods' structure does not contain an explicit way of building series decomposition.

To rectify the listed disadvantages, we have developed a new method called *tensor SSA* (tSSA). It has only two adjustable parameters. The tSSA extends an SSA method [12] for the multivariate time series and implies using dynamical systems theory.

The main task of the paper is to find a shared *phase subspace* [13, 14] for all given time series. This is a low-dimensional linear subspace where all the system's trajectories can be mapped into isomorphically. Finding it enables us to obtain phase representation of observed signals. Fig. 1 illustrates this idea. We use a single time series to build phase space and the specific trajectory of the dynamical system that generated the given series. The forecast then is found upon the fact that the continuation of the trajectory lies in the very same phase space. Partitioning basis vectors of this space into several groups builds us the decomposition of the series. The paper develops named conceptions in the next sections.

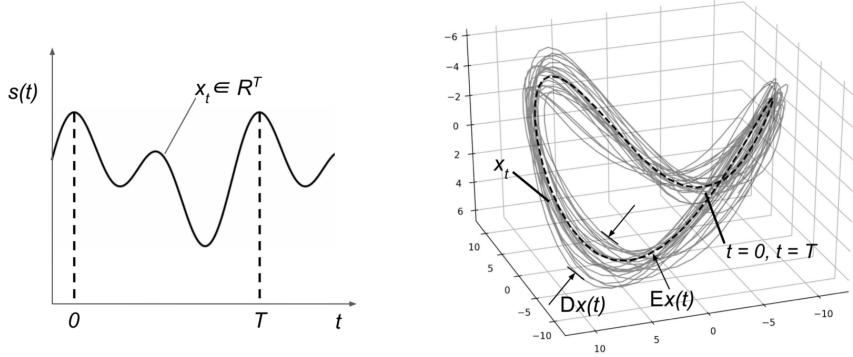


Figure 1 Time series's phase trajectory visualization

Our method captures autocorrelation and multilinearity of the time series via tensor data representation. *Canonical polyadic decomposition* (CPD) is used to build the phase space of the series. Another solution is to use matrix data representation and matrix decompositions. Such method is called matrix SSA (mSSA) [15]. It is the second extension of the SSA method for the forecast and the decomposition of the multivariate time series. The tSSA is compared with the mSSA in the computational experiment section.

The rest of the paper covers the theory and application of our method to real data. Firstly, the dynamical systems model is introduced and the problem of basis search in the phase space is stated and resolved. These results enable us to propose a way of decomposing the series and to make the forecast. Simultaneously, features of obtained solutions are examined and two associated theorems are formulated. Finally, tSSA and mentioned methods are applied to two datasets: electricity consumption and accelerometer/gyroscope observations. We obtain their predictions and additive decompositions. For the latter, a special quality metric is introduced.

2 Problem statement

Suppose a *dynamical system* with the phase space X , evolution function f and the initial state \mathbf{y}_0 . Time space is discrete

$$\mathbf{y}(t+1) = f(\mathbf{y}(t)) \quad t \in \mathbb{N},$$

$$\mathbf{y}(0) = \mathbf{y}_0.$$

Generally, X is a high-dimensional smooth manifold. Then unknown mapping $\phi : X \rightarrow \mathbb{R}^m$ takes each trajectory point $\mathbf{y}(t)$ to the multivariate time series realization \mathbf{x}_t

$$\phi(\mathbf{y}(t)) = \mathbf{x}_t. \text{ Coordinateley: } \begin{cases} \phi_1(\mathbf{y}(t)) = x_1(t), \\ \dots \\ \phi_m(\mathbf{y}(t)) = x_m(t). \end{cases}$$

It is assumed that the trajectories $\mathbf{y}(t)$ belong to the smaller dimension manifold $M \subset X$. Now the problem is to find an embedding of the M into \mathbb{R}^L for some L . The second problem is to build a basis in the image of the embedding. Having done that, the initial dynamical system will be described in terms of the standard linear space. The same will hold to all time series $x_i(t)$.

2.1 Single-variate series case

The tesnor SSA relies on the SSA method and the Takens theorem [16]. The theorem provides the needed embedding in the single-variate series case. Any point $\mathbf{y}(t) \in M$ corresponds to the following vector:

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

It is called a *delay vector* at time t and is denoted as $\overleftarrow{\mathbf{x}}_t$. The vector's dimension L must satisfy condition $L > 2 \cdot \dim(M)$. The function $\phi(\cdot)$ must satisfy several regularity conditions. They are omitted here and considered fulfilled.

Now the time series $x(t)$ with the length N gives $N - L + 1$ delay vectors. Therefore image of the embedding is $\text{Lin}(\{\overleftarrow{\mathbf{x}}_t\})$. It represents the phase space of the series and must be low dimensional. This means $\text{Lin}(\{\overleftarrow{\mathbf{x}}_t\}) \subset \mathbb{R}^L$. Finally, the orthonormal basis is an U -component of the Singular value decomposition (SVD) on a *trajectory matrix* \mathbf{H}_x . It is a stacking of the delay vectors

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

2.2 Multivariate series case and the tSSA method

Now there given multivariate time series. Let m be their number. Hence the $\phi(\cdot)$ is a multidimensional map. This means that instead of the single delay vector we have m of them at any time t . Let $\overleftarrow{\mathbf{X}}_t$ be a matrix $(\overleftarrow{\mathbf{x}}_{1_t} \dots \overleftarrow{\mathbf{x}}_{m_t})$ called *delay matrix*. Then, the embedding image is a linear span of all $\overleftarrow{\mathbf{X}}_t$. It is also called the *shared phase space* of the multivariate time series.

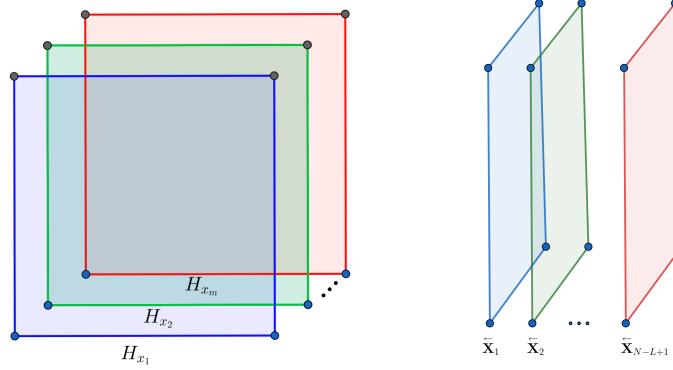


Figure 2 Two views on trajectory tensor. The left is in terms of signals' trajectory matrices $\{x_i(t)\}_{i=1}^m$. The right is in terms of delay matrices.

Similarly to the trajectory matrix we introduce *trajectory tensor* denoted by \mathbf{T} . It is the delay matrices stacked along the second dimension of the tensor, Fig. 2 (left). Equivalent way of constructing the tensor is the following. The \mathbf{T} is the trajectory matrices \mathbf{H}_{x_i} stacked along the third dimension, Fig. 2 (right). Therefore the shared phase space of the series is equal to a linear span of all \mathbf{H}_{x_i} . As discussed in the Sec. 2.1, each trajectory matrix also defines the phase space for the corresponding time series $x_i(t)$. Finally, we give a

Definition. The multivariate time series are said to be *interdependent* if they share the common phase space with the same basis.

Now assuming observed series interdependence it follows that each \mathbf{H}_{x_i} has a matrix factorization with the same set of factors. Since the trajectory matrices constructs the trajectory tensor, it has a low-dimensional representation. Using these two corollaries we apply the CP-decomposition to the \mathbf{T} and obtain $\mathbf{T} = \sum_{i=1}^r \mathbf{a}_i \otimes \mathbf{b}_i \otimes \mathbf{c}_i$. The CPD is defined by the tensor rank r and a set of vectors. The vectors are usually composed in full-rank matrices $A = [\mathbf{a}_1 \dots \mathbf{a}_r]$, $B = [\mathbf{b}_1 \dots \mathbf{b}_r]$, $C = [\mathbf{c}_1 \dots \mathbf{c}_r]$. Denote by $\boldsymbol{\sigma}_{x_k}$ the k -th row of the C . Finally, with respect to the third-dimension slices of the \mathbf{T} , CPD goes as

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

The CPD exists for any tensor. In case of the trajectory tensor three properties must hold.

Theorem 1. Suppose that the given multivariate time series are interdependent. Then

1. Any $\boldsymbol{\sigma}_{x_k}$ is a non-zero vector
2. The tensor rank $r \leq L$
3. Transposition of the trajectory matrix is a trajectory matrix. The shared phase space for the transposed matrices is the $\text{Lin}(\{\mathbf{b}_i\})$

Proof. 1. Suppose some vector has a zero element at the position j and the other is non-zero in this position. Then two trajectory matrices factorize with a different set of factors. That violates the definition of the series interdependence. Another

case is that all σ_{x_k} have a zero element at the position j . Then the matrix C has a zero row and is not full-ranked. It contradicts to the CPD definition.

2. As shown in the Sec. 2.1 each time series has its phase space dimension $\leq L$. All delay vectors of every series also lie in the shared phase space $\text{Lin}(\{\mathbf{a}_i\})$. Therefore its dimensionality can not become greater.
3. It is obvious that rows of the trajectory matrices are the delay vectors of size $N - L + 1$. Transpose all equalities in the (1). Now it is clear that rows of any \mathbf{H}_{x_i} belongs to the $\text{Lin}(\{\mathbf{b}_i\})$. This is the shared phase space for the new delay vectors.

□

2.3 Time series decomposition

The series are decomposed into additive components using the following conception: *factorization of the trajectory matrix \mathbf{H}_{x_k} defines the decomposition of the related time series*. This idea comes from the SSA method. Since all \mathbf{H}_{x_k} have a similar factorization, each time series is decomposed separately.

Trajectory matrices of the series are *hankel* by their definition. That means equal elements along each anti-diagonal of the matrix. It is trivial that any time series of length N bijectively corresponds to a hankel matrix of size $L \times (N - L + 1)$.

Now, we describe the decomposition algorithm. First, the factors of the \mathbf{H}_{x_k} are arbitrary partitioned into s groups. The s is also a chosen number. Then all factors within each partition are summed up. As a result we obtain matrices C_1, \dots, C_s . If they are all hankel then each C matrix corresponds to the time series component of the $x_i(t)$. However, it is almost infeasible even with the trivial series [12]. Therefore every C matrix is additionally *hankelized*. This operator averages each matrix's anti-diagonals so the matrix becomes hankel. Denote it as $\text{Hankel}(\cdot)$. Finally, the algorithm can be written as a chain of identical transformations

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

Here $\mathbb{I}_1 \sqcup \dots \sqcup \mathbb{I}_s = \{1, \dots, r\}$ are chosen groups of indices. The fourth equality requires proof. But first, consider a hankel operator's property:

Lemma 1. *The $\text{Hankel}(\cdot)$ is a linear operator.*

Proof. Having matrices A and B with the same size, take their elements from any anti-diagonal. Denote them by a_1, \dots, a_n and b_1, \dots, b_n . Then this anti-diagonal in $\text{Hankel}(A + B)$ is $\frac{1}{n} \sum_{i=1}^n (a_i + b_i) = \frac{1}{n} \sum_{i=1}^n a_i + \frac{1}{n} \sum_{i=1}^n b_i$. It exactly results in a sum of $\text{Hankel}(A)$ and $\text{Hankel}(B)$ anti-diagonals.

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

Now recall that the \mathbf{H}_{x_k} is hankel. Combining it with the previous lemma, apply $\text{Hankel}(\cdot)$ to the $\mathbf{H}_{x_k} = C_1 + \dots + C_s$. This proofs the fourth equality at (2). Therefore, we have shown the correctness of the decomposition algorithm.

2.4 Optimal decomposition problem

To eliminate ambiguity in the partitioning of the \mathbf{H}_{x_k} factors, we restrict C_j matrices to be hankel. Thus, the fourth step of the decomposition algorithm (2) becomes unnecessary. In addition, such partitioning completely fulfills the decomposition concept.

Now we formulate the decomposition problem as a system of matrix equations. Take the first equality from the (2) and apply the hankel operator to the both sides. Recall that the \mathbf{H}_{x_k} is hankel. Also take the very same equality as it is. We obtain

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

Then subtract the second from the first and denote by H_i a $\boldsymbol{\sigma}_{x_k}(i) \cdot \mathbf{a}_i \mathbf{b}_i^\top - \text{Hankel}(\boldsymbol{\sigma}_{x_k}(i) \cdot \mathbf{a}_i \mathbf{b}_i^\top)$. It is called the *hankel residual matrix* of the i -th factor. Therefore we get

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

Finally, the condition on the C_i to be hankel is reformulated as the residual matrices sum up to zero within each partition group. The system of matrix equations is the following

$$\sum_{k \in \mathbb{I}_i} H_k = 0 \quad \forall i \in 1, \dots, s.$$

Without loss of generality, we set s equals two. If this problem is solved then the resulting groups of the factors can be further partitioned. Hence any number of the time series components can be obtained. Moreover, for any \mathbf{H}_{x_k} its residual matrices are linearly dependent as follows from the (3). Thus, the partition group for the H_r is uniquely defined if all other residuals are already partitioned. This matrix can be discarded. We also assume all H_j to be non-zero matrices. Otherwise, the zero residual matrix alone and all the rest make the desired partition on two groups. This assumption implies each of the groups to have at least two residual matrices. Finally, for any H_j matrix introduce an indicator-variable $\beta_j \in \{0, 1\}$. It shows what group the residual matrix belongs to. Denote by $\boldsymbol{\beta}$ a vector $(\beta_1 \dots \beta_{r-1})^\top$. The decomposition problem is to find $\boldsymbol{\beta}$ such that

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

The [12] shows that even for trivial time series such a problem is infeasible. Now we transform it into a feasible one. First, vectorize each H_i and stack resulting vectors into the matrix denoted by Λ . Then the first equation in the (4) is equivalent to the $\Lambda\beta = 0$. Second, change the problem of solving the system of equations to an optimization problem. Let's find such β that delivers minimum to the norm of the $\Lambda\beta$. The changed problem is

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

Call solution of the (5) the *optimal series decomposition*. The problem itself is an Integer Least Squares (ILS) problem, which is proved to be NP-hard [17]. The final result of the section is

Theorem 2. *The optimal series decomposition is an NP-hard problem.*

In spite of the complexity, methods and heuristics exist to reduce the ILS to computationally effective problems [18].

2.5 Time series forecasting

Using the build shared phase space each time series is forecast individually. We make a one-step forecast to estimate $x(N + 1)$. Phase trajectory of the series is a sequence of the delay vectors $\{\overleftarrow{\mathbf{x}}_t\}$. The whole trajectory belongs to the phase space $\text{Lin}(\{\mathbf{a}_i\})$. Continuation of the trajectory is the $\overleftarrow{\mathbf{x}}_{N+1}$. Therefore $\overleftarrow{\mathbf{x}}_{N+1} \in \text{Lin}(\{\mathbf{a}_i\})$. At the same

time $x(N+1)$ is the last component of the $\overleftarrow{\mathbf{x}}_{N+1}$. Now, recall the A matrix introduced in the Sec. 2.2. The system of linear equation to find $x(N+1)$ is

$$\begin{aligned} \overleftarrow{\mathbf{x}}_{N+1} = A\boldsymbol{\lambda} \text{ or equally } & \begin{cases} \mathbf{x}_{kn} = A_{kn}\boldsymbol{\lambda} \\ x(N+1) = \mathbf{a}_{pr}^\top \boldsymbol{\lambda} \end{cases}, \text{ where} \\ & A = \left(\frac{A_{kn}}{\mathbf{a}_{pr}^\top} \right), \\ & \overleftarrow{\mathbf{x}}_{N+1} = (\mathbf{x}_{kn} \ x(N+1))^\top. \end{aligned} \quad (6)$$

The delay vector and the matrix are broken into two blocks associated with the *known* and *predicted* part of the series. The forecast is made using the second equation in the (6). Therefore vector $\boldsymbol{\lambda} \in \mathbb{R}^r$ is to be found from the first equation in the (6). It is a system of linear equations with the matrix A_{kn} . The matrix has a rank at least $r - 1$ since A matrix is full-ranked. However we assume A_{kn} to have rank r too. As experiments show this is not burdensome. In addition, the linear system is overdetermined since $r \leq L$ (see Sec. 2.2). Therefore its solution is given by the least-squares formula $\boldsymbol{\lambda} = (A_{kn}^T A_{kn})^{-1} A_{kn}^T \mathbf{x}_{kn}$. After replacing it in (6) we obtain the forecast formula

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

Denote by the \mathbf{d} a vector $\mathbf{a}_{pr}^\top (A_{kn}^T A_{kn})^{-1} A_{kn}^T$. Once computed, it can be reused to make the forecast for any steps further. Now rewrite the (7) in terms of the $x(t)$:

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

This expression reveals the property of the forecast sequence $\{\hat{x}(N+i)\}_{i=1}^\infty$.

Theorem 3. *The model of the tSSA's forecast is autoregressive AR($L - 1$). The coefficients d_i found in the (7) completely define behavior of the forecast sequence.*

3 Computational experiment

In the current section we obtain the decomposition and the forecast of multivariate time series. We also compare the tSSA with the other methods: the mSSA, the VAR, and the RNN. The latter two are only for the forecast. Moreover, to measure quality of the decomposition we introduce special metrics. They are motivated by the corresponding analysis in the Sec. 2.3.

Definition. The *absolute hankel error* of the matrix M is

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

Definition. The *relative hankel error* of the matrix M is

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

The AHE is proportional to the total variance of the matrix's anti-diagonals. The RHE is AHE's more interpretable modification. In the Sec. 2.3 matrices C_i were introduced. They correspond to the time series components. In the Sec. 2.4 the optimal decomposition was introduced. It is basically minimizing of the AHE for the C_i matrices. Hence, the AHE and the RHE are computed for the C_i in the further experiments. Denote by $\overline{\text{RHE}}_{ts}$ the mean RHE across all C_i of the time series ts . Denote by $\overline{\text{RHE}}$ the mean $\overline{\text{RHE}}_{ts}$ across all given time series.

To measure forecast quality for the time series ts we use standard metrics: the *mean squared error* MSE_{ts} and the *mean absolute percentage error* MAPE_{ts} . Denote by $\overline{\text{MSE}}$ and $\overline{\text{MAPE}}$ their mean across all given time series.

Two sets of data are involved in the experiment. First, electricity consumption and electricity price time series (Fig. 3). Second, inertial unit measurements; three time

series are from an accelerometer and three from a gyroscope [19] (Fig. 4). Walking movements are detected here. We assume the series' interdependence (see Sec. 2.2) based on the economical/physical premises. The series have a length of $\approx 3 \cdot 10^3$. About 20% of them are deferred test samples for the forecast quality evaluation.

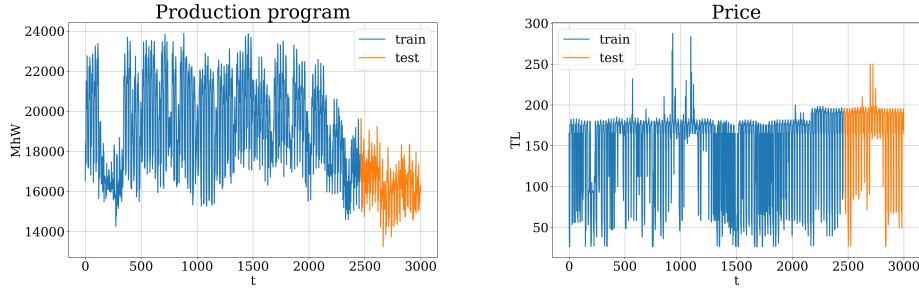


Figure 3 Time series for electricity consumption and its price

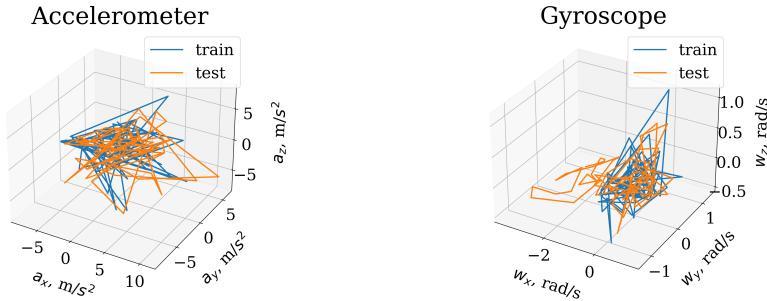


Figure 4 Time series for linear and angular accelerations

Now we describe some computational features of the tSSA application. First, tensor rank computation is NP-hard [20]. Its specific number is explicitly indicated whenever needed. Second, an algorithm for CPD computation is the ALS [21]. Since it has an approximation error, the (1) is not precise on practice. Therefore, computed forecast and decomposition are not precise too. The approximation errors are specified whenever needed. Third, the ILS problem (5) must be solved to find the series

decomposition. But the involved matrix Λ has a large row dimension. To shorten computation, only several hundred rows are left. However, it is still much greater than the size of the sought β vector. Finally the L value (see Sec. 2.1) is 500 for the electricity data and 1000 for the inertial unit data. This parameter also has a meaning for the other models. For the RNN it is the length of the initial series input to further make the forecast. For the VAR it is a maximal lag number [4].

3.1 Data availability

Electricity consumption dataset is available at <https://sourceforge.net/p/mvr/code/HEAD/tree/data/TurkElectricityConsumption.csv>. Accelerometry data is available at <https://data.mendeley.com/datasets/45f952y38r/5>.

3.2 Results and discussion

We begin with the tSSA's forecast results. First, relationship between the forecast metrics and the trajectory tensor's rank is depicted in the Fig. 5 and 6. We see that the forecast quality decreases with the greater ranks for both metrics. Furthermore, *overtraining effect* is present as the MAPE metric increases after reaching a minimum. Nonetheless, the CPD's approximation error decreases monotonously, see Fig. 7.

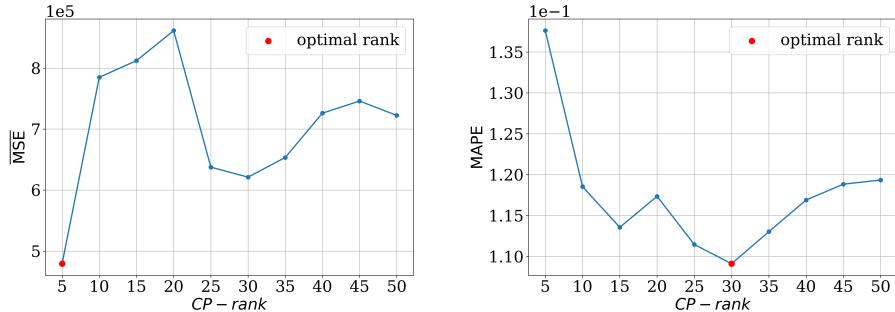


Figure 5 $\overline{\text{MSE}}$ and $\overline{\text{MAPE}}$ metrics for the tSSA forecast depending on the CPD rank. An optimal point is marked with red. Electricity data.

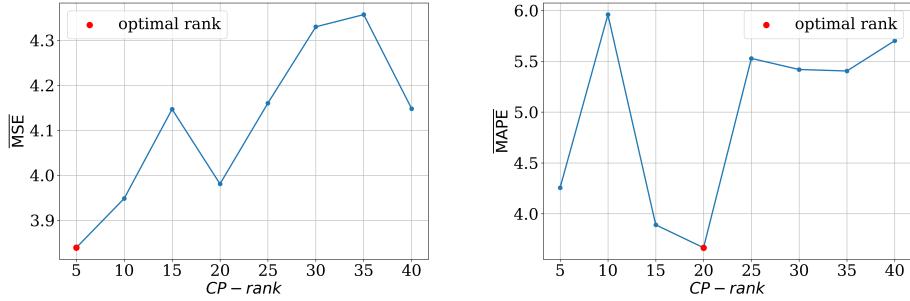


Figure 6 $\overline{\text{MSE}}$ and $\overline{\text{MAPE}}$ metrics for the tSSA forecast depending on the CPD rank. An optimal point is marked with red. Inertial unit data.

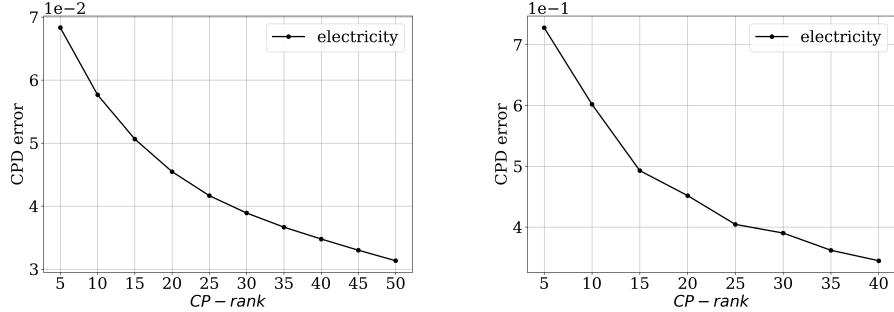


Figure 7 Relative CPD approximation errors depending on the CPD rank. The left is for the electricity data, the right is for the inertial unit data.

The best tSSA forecast is shown in the Fig. 8 and 9. For electricity, the CPD rank is rather big. Therefore, the order of the forecast model (see Sec. 2.5) is high. The resulting forecast approximates test data quite accurately. On the other hand, big rank for the inertial unit data lead to the unstable and unbound forecast. With the smaller rank, model is more accurate. However, the forecast model's order is decreased. Consequently, the build forecast approximates only mean values.

The forecast metrics for all models are in the Tab. 1 and 2. Our method showed the best results in most cases. Close values were obtained with the mSSA. The VAR model appeared unstable for the chosen forecast test data. The RNN was able to learn only the constant function from the electricity data. But with the greater L for the inertial unit data, the RNN performed better.

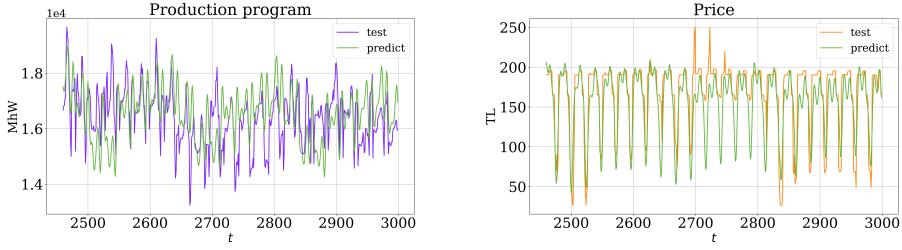


Figure 8 tSSA forecast for the electricity data. The CPD rank = 30

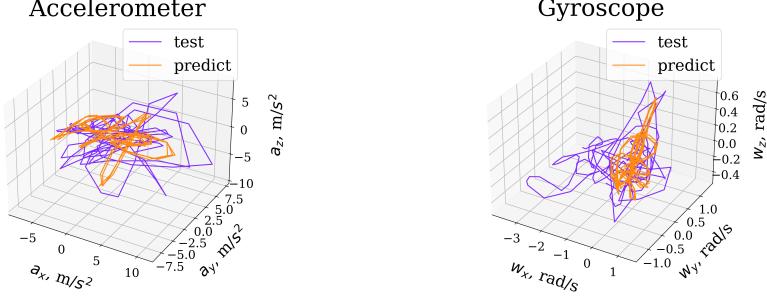


Figure 9 tSSA forecast for the inertial unit data. The CPD rank = 20

Table 1 Prediction quality of the models on the 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 results. The Fig. 10 illustrates relationship between the quality metrics and the CPD rank. In both cases the $\overline{\text{RHE}}$ rapidly reaches a minimum and then does not change or even increase. Recalling the same result for the forecast, we can conclude that the tSSA has the greatest generalization capacity for the small tensor ranks. At the same time, the rank is a value of the shared phase space dimension (see Sec. 2.2). Therefore, the tSSA finds a low dimensional data representation as was indicated in the Sec. 2.

Table 2 Prediction quality of the models on the inertial unit data

	<i>tSSA</i>	<i>mSSA</i>	<i>VAR</i>	<i>RNN</i>
$\overline{\text{MSE}}_{\text{Accel}}$	7.351	6.980	8.108	6.604
$\overline{\text{MSE}}_{\text{Gyro}}$	0.610	0.636	0.631	0.639
$\overline{\text{MSE}}$	3.981	3.808	4.370	3.622
$\overline{\text{MAPE}}_{\text{Accel}}$	3.558	3.516	3.370	1.747
$\overline{\text{MAPE}}_{\text{Gyro}}$	3.773	3.943	10.427	5.641
$\overline{\text{MAPE}}$	3.666	3.730	6.899	3.694

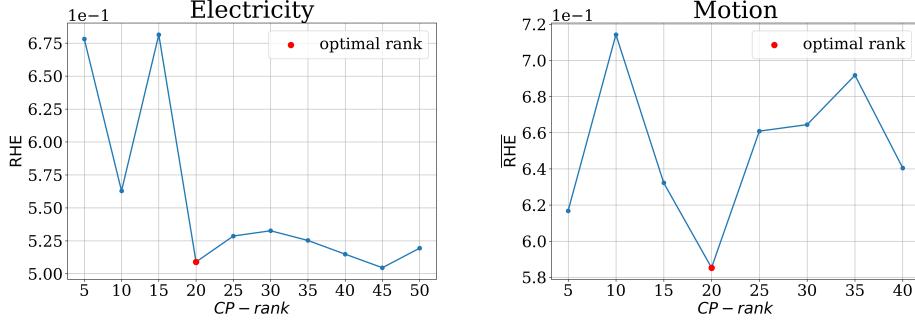


Figure 10 Relationship between the $\overline{\text{RHE}}$ metrics and the CPD rank. The left is for the electricity data, the right is for the inertial unit data. An optimal point is marked with red

The decomposition on two components for the electricity consumption and its price is illustrated in the Fig. 11. The components are almost identical to the bias and scale factors. This is attributed to an effect of having the *shared* phase space between the time series. Two-component decomposition for the accelerometer is shown in the Fig. 12 and for the gyroscope in the Fig. 13. The same effect is seen for the second components here.

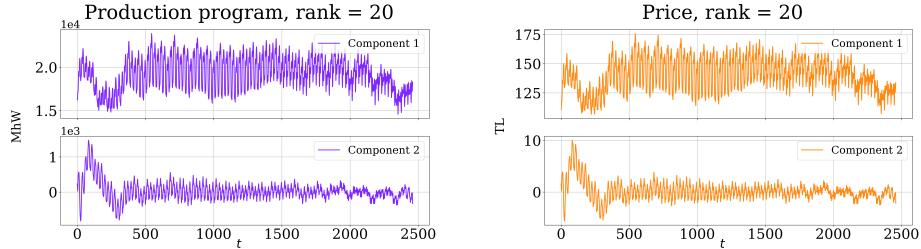


Figure 11 tSSA decomposition for the electricity data. CPD rank = 20

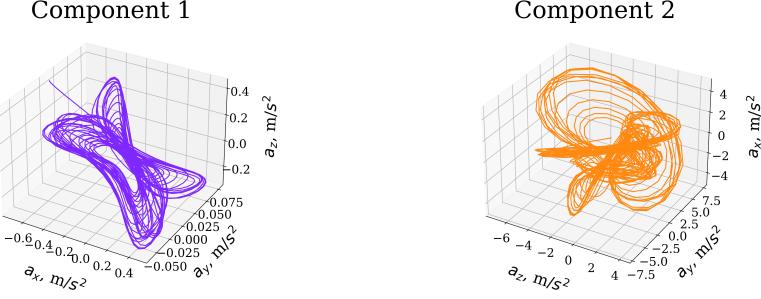


Figure 12 tSSA decomposition for the accelerometer data. CPD rank = 10

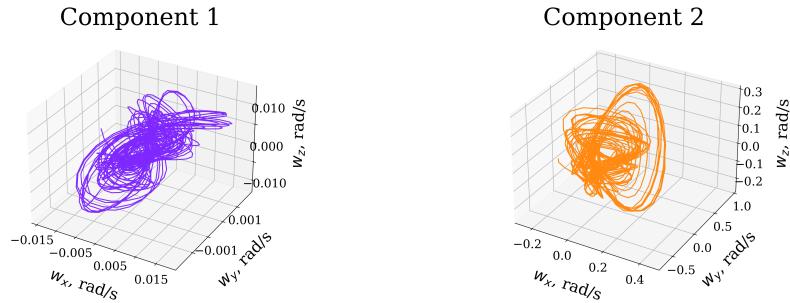


Figure 13 tSSA decomposition for the gyroscope data. CPD rank = 10

We also present the mSSA's decomposition for the electricity data in the Fig. 14 and for the inertial unit data in the Fig. 15, 16. The decomposition here is similar to the tSSA and connected with the factors partitioning. Several heuristics exist to make the partitioning manually [12]. As shown in the Tab. 3 and 4, the decomposition metrics are greater for that heuristics then for the tSSA. Number of the mSSA's components is not limited. However, such results are not always guaranteed. Automation here is hindered. Moreover, the mSSA's components for the inertial unit data are trivial and do not reflect complex dynamics of the mechanical motion. To summarize, the complexity of the tSSA decomposition does not allow for building many components but guarantees their optimal quality in terms of the AHE and RHE.

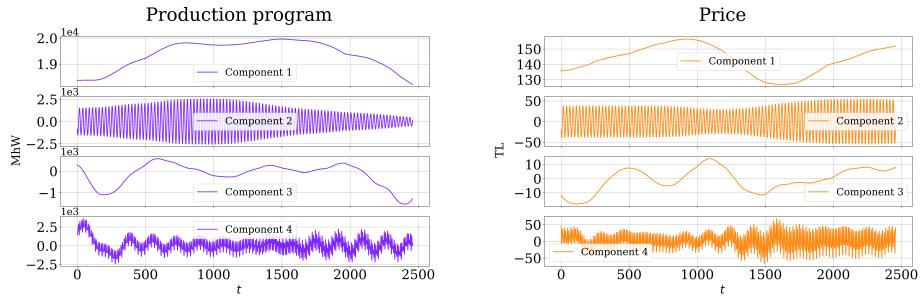


Figure 14 mSSA decomposition for the electricity data

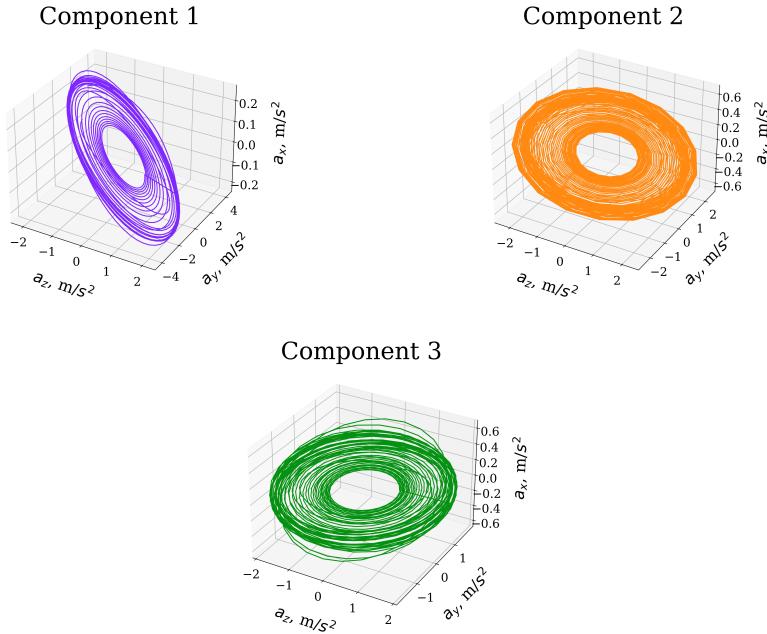


Figure 15 mSSA decomposition for the accelerometer data

Table 3 Decomposition quality of the models for the electricity data

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

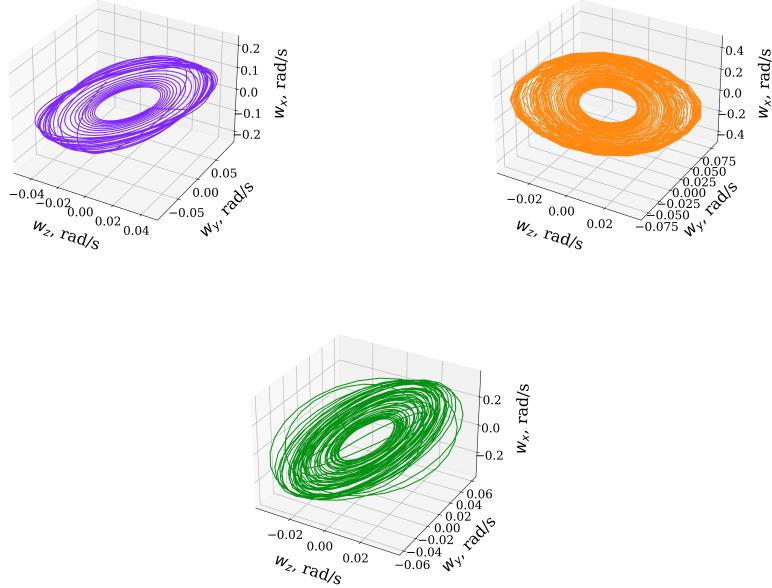


Figure 16 mSSA decomposition for the gyroscope data

Table 4 Decomposition quality of the models for the inertial unit data

	tSSA	mSSA
RHE _{Accel}	0.438	0.394
RHE _{Gyro}	0.732	0.468
RHE	0.585	0.431

4 Conclusion

Proposed tensor method is devoted to the forecast and the additive decomposition of the interdependent multivariate time series. It is based on dynamical systems theory and explicitly build the shared phase space of the series. It also exploits multilinearity of the data via tensor data representation. The method has only two adjustable parameters and require the CPD computation of the trajectory tensor. Then, the forecast comes down to the vector's scalar product. The decomposition comes down to matrix factorization and factor's partitioning. Finally, the computational experiment

showed tSSA's high forecast and decomposition quality. The method also reduced dimensionality of the given data.

A tSSA's disadvantage is an NP-hardness of the optimal series decomposition. Relaxation of this problem or finding another principle to build additive components are the major directions for further method improvement. Another future research task is the extension of the dynamical systems approach and proposing a new way of building the shared phase space.

References

- [1] Enders, W., Sons, J.W.: Applied Econometric Time Series. Wiley series in probability and statistics. Wiley, New York (2010).
<https://books.google.ru/books?id=vzJ0CgAAQBAJ>
- [2] Dagum, E., Bianconcini, S.: Seasonal Adjustment Methods and Real Time Trend-Cycle Estimation, p. 283. Springer, Switzerland (2016). <https://doi.org/10.1007/978-3-319-31822-6>
- [3] Cleveland, R.B., Cleveland, W.S., McRae, J.E., Terpenning, I.: Stl: A seasonal-trend decomposition procedure based on loess (with discussion). Journal of Official Statistics **6**, 3–73 (1990)
- [4] Hyndman, R.J., Athanasopoulos, G.: Forecasting: Principles and Practice. OTexts, Australia (2021). <https://books.google.ru/books?id=gZB-zgEACAAJ>
- [5] Box, G.E.P., Jenkins, G.M.: Time Series Analysis: Forecasting and Control. Holden-Day series in time series analysis and digital processing. Holden-Day, Minnesota (1970). <https://books.google.ru/books?id=5BVfnXaq03oC>
- [6] Motrenko, A., Simchuk, E., Khairullin, R., Inyakin, A., Kashirin, D., Strijov, V.: Continuous physical activity recognition for intelligent labour monitoring.

Multimedia Tools and Applications **81**(4), 4877–4895 (2022) <https://doi.org/10.1007/s11042-021-11288-y>

- [7] Volterra, V., Brelot, M.: Leçons sur la Théorie Mathématique de la Lutte pour la vie. Cahiers Scientifiques. Fasc Series. Gauthier-Villars et cie., Michigan (1931). <https://books.google.ru/books?id=12rQAAAAMAAJ>
- [8] Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation **9**(8), 1735–1780 (1997) <https://doi.org/10.1162/neco.1997.9.8.1735>
- [9] Tealab, A.: Time series forecasting using artificial neural networks methodologies: A systematic review. Future Computing and Informatics Journal **3**(2), 334–340 (2018) <https://doi.org/10.1016/j.fcij.2018.10.003>
- [10] Asteriou, D., Hall, S.: Vector Autoregressive (VAR) Models and Causality Tests, pp. 333–346 (2016). https://doi.org/10.1057/978-1-137-41547-9_15
- [11] Zellner, A.: An efficient method of estimating seemingly unrelated regressions and tests for aggregation bias. Journal of the American Statistical Association **57**(298), 348–368 (1962) <https://doi.org/10.1080/01621459.1962.10480664>
- [12] Stepanov, D., Golyandina, N.: SSA-based approaches to analysis and forecast of multidimensional time series, pp. 293–298 (2005)
- [13] Krim, H., Viberg, M.: Two decades of array signal processing research: The parametric approach. Signal Processing Magazine, IEEE **13**, 67–94 (1996) <https://doi.org/10.1109/79.526899>
- [14] Ignatov, A.D., Strijov, V.V.: Human activity recognition using quasiperiodic time series collected from a single tri-axial accelerometer. Multimedia tools and applications **75**, 7257–7270 (2016) <https://doi.org/10.1007/s11042-015-2643-0>

- [15] Hassani, H., Mahmoudvand, R.: Multivariate singular spectrum analysis: A general view and new vector forecasting approach. International Journal of Energy and Statistics **01** (2013) <https://doi.org/10.1142/S2335680413500051>
- [16] Takens, F.: Detecting Strange Attractors in Turbulence. In: Rand, D., Young, L.-S. (eds.) Dynamical Systems and Turbulence, Warwick 1980. Lecture Notes in Mathematics, vol. 898, pp. 366–381. Springer, Berlin (1981). Chap. 21. <https://doi.org/10.1007/bfb0091924>
- [17] Emde-Boas, P.: Another NP-complete Partition Problem and the Complexity of Computing Short Vectors in a Lattice. Report. Department of Mathematics. University of Amsterdam. Department, Univ., Amsterdam (1981). <https://books.google.ru/books?id=tCQiHQAAQAAJ>
- [18] Grafarend, E., Zwanzig, S., Awange, J.: Integer Least Squares, pp. 579–603. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-94598-5_16
- [19] Nahid, A.-A., Sikder, N., Rafi, I.: KU-HAR: An Open Dataset for Human Activity Recognition. Mendeley Data (2021). <https://doi.org/10.17632/45f952y38r.5>
- [20] Håstad, J.: Tensor rank is np-complete. Journal of Algorithms **11**(4), 644–654 (1990) [https://doi.org/10.1016/0196-6774\(90\)90014-6](https://doi.org/10.1016/0196-6774(90)90014-6)
- [21] Kolda, T., Bader, B.: Tensor decompositions and applications. SIAM Review **51**, 455–500 (2009) <https://doi.org/10.1137/07070111X>