View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by DSpace at Tartu University Library

UNIVERSITY OF TARTU

Faculty of Science and Technology

Institute of Mathematics and Statistics

Aytan Osmanzade

Singular spectrum analysis forecasting for financial time series

Financial Mathematics

Master Thesis (15 ECTS)

Supervisor: Peep Miidla

Tartu 2017

Singulaarse spektraalanalüüsi meetod aegridade prognoosimiseks

Lühikokkuvõte. Singulaarse spektraalanalüüsi meetod (SSA) on suhteliselt uus mitteparameetrilne andmepõhine meetod aegridade analüüsiks, aga see on rakendust leidnud ka muude ülesannete korral eri valdkondades, üldisemalt – algülesande dimensioonide vähendamiseks. Käesolevas magistritöös antakse ülevaade SSA põhiskeemist ja prognoosimismeetodist aegridade korral. SSA põhialgoritm koosneb erinevatest etappidest: dekompositsioon, rekonstruktsioon ja prognoosimine. Töö on jaotatud viieks peatükiks, sisaldades kirjanduse loetelu ja lisad. Peatükis 4 esitatakse autori poolt rakenduspaketiga Matlab tehtud arvutuseksperimentide tulemused finantsturu ühe aegrea jaoks. Tööle on lisatud programmide tekstid, algandmed ja arvutustulemused.

Märksõnad: Singulaarse spektraalanalüüsi meetod (SSA), finantsaegread, prognoosimine, sulgemishinnad.

CERCS teaduseriala: P160 Statistika, operatsioonianalüüs, programmeerimine, finants- ja kindlustusmatemaatika.

Singular spectrum analysis forecasting for financial time series

Abstract. SSA is a relatively new non-parametric data-driven technique in time series analysis, has been developed and applied to many practical problems across different fields. This paper focuses on the technique of Singular Spectrum Analysis (SSA), its application for financial time series, and also represents results of numerical experiments done by author. The main algorithm of SSA consists of two complementary stages: decomposition and reconstruction; both stages include two separate steps. The performance of the SSA technique is assessed by applying it to the close prices of "AS Tallink Grupp" stock. Results in this work are obtained from creation trajectory matrix of given time series and finding eigenvalues and eigenvectors; construction of the principal and reconstructed components of the time series; applying forecasting algorithm to the time series; interpretation of obtained results. In this thesis for numerical experiments we use the software Matlab.

Keywords: Singular Spectrum Analysis, Financial Time Series, Forecasting, Close Prices.

CERCS research specialisation: P160 Statistics, operations research, programming, actuarial mathematics.

DEDICATION

To my parents

and

my brother

who always loved, cared and supported me.

ACKNOWLEDGMENTS

In first place I want to thank my supervisor, Dr Peep Miidla who gave me the guidelines and always supported me on my thesis. I have gained a wide range of new information and valuable experiences while working with him. I appreciate his dedication and assistance towards my work, and provide useful feedback and suggestions. Moreover, I am grateful him for his valuable advice, comments, encouragement and patience.

It is my pleasure to acknowledge all the professors, lecturers, other office staffs and my classmates of Institute of Mathematical Statistics, the University of Tartu, specially Dr Kalev Pärna, PhD Raul Kangro, MSc Annika Krutto and PhD Meelis Käärik for their support during my studies.

I would like to thank board of directors and team of the Azerpost LLC, specially Kazimov Subhan, Suleymanov Nazim and Jafarov Farid that they believed in me and supported my study in Estonia.

I want to thank my family who have been with me during this journey. My parents, Osmanov Ali and Farzaliyeva Rahima, have provided a great support during my studies. They encouraged me to travel to Estonia to pursue this degree and to achieve this goal. I also wish to thank to my brother Osmanzade Nusrat, for his help in reviewing my work, for his valuable advises and for being there for me when I needed.

Finally, I want to say thanks to God for all His guidance and wisdom.

Contents

1. INTRODUCTION

A time series insures information about the physical, biological, socioeconomic or financial systems that produced it. The purpose time series analysis based the pattern of time series, to determine some of the main properties, to understand how the time series has behaved in the past. It can then help to understand and predict the system's future behaviour [15]. There are many well-known decomposition methods which is used in time series analysis. In this paper we study Singular Spectrum Analysis and it's application in financial time series.

In recent years SSA, a comparatively new but powerful technique in time series analysis, has been researched and applied to many practical problems in different fields. SSA is non-parametric time series method which decomposes, reconstructs and forecasts time series. It incorporates tools from time series analysis, multivariate statistics, dynamical systems and signal processing [17]. The basic SSA method primarily involves two stages: decomposition and reconstruction and both stages contain two separate steps. The decomposition stage consists two steps: embedding and singular value decomposition (SVD); the reconstruction stage: grouping and diagonal averaging. These two stages make up the basic SSA algorithm [16]. The structure of SSA algorithm is as follows. First, a one-dimensional time series is converted into a higher dimension matrix is called the trajectory matrix. The dimension of the trajectory matrix is called the window length. In the second step SVD is applied to the trajectory matrix and eigenvalues and eigenvectors are found. The next step is the grouping step which involves splitting the elementary matrices into several groups and summing the matrices in each group. By taking the average along the diagonals of each group we get reconstructed components and combining them into one time series we obtain the approximated time series of the initial series [16]. Additionally to these stages, an important advantage of SSA is that it allows, after reconstruction of the time series under study, to produce forecasts for the reconstructed components which is called SSA forecasting algorithm.

The purpose of this work is to study and understand the SSA method, SSA forecasting algorithm, to make numerical experiments on financial time series and to compare its performance with different sampling SSA parameters.

The work is made up of five main sections including the introduction and conclusion. In section 2, we provide a review of the main linear algebra tools and brief summary of LU decomposition, eigenvalues and eigenvectors, spectral and singular decomposition. Section 3, involves theoretical study about time series, basic SSA method, forecasting algorithm, choice of SSA parameters and forecast accuracy. In section 4, we apply SSA to a financial time series- the close prices of "AS Tallink Grupp" stock taken from Yahoo! Finance and make numerical experiments on chosen data. In this section first reconstructed components are built and shown that sum all reconstructed components gives initial time series. Next we use the reconstructed components for forecasting new data points. One of main tasks in this experiments to compare forecasting results which is obtained with different window length and number of components and to determine a suitable sampling for them. We discuss the conclusion and future work in sections 5. The numerical experiments in this study were done with Matlab which has been studied during this work.

2. LINEAR ALGEBRA TOOLS

2.1. LU Decomposition

LU decomposition is a method of factorization of square matrix A. It will yield a product of a lower triangular matrix (L) and an upper triangular matrix (U). For example, given $n \times n$ matrix A the decomposition is [13, 14]

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & \dots & 0 \\ l_{21} & l_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \dots & l_{nn} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ 0 & u_{22} & \dots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & u_{nn} \end{bmatrix}.$$

Let us consider the LU factorization for 3×3 matrix:

$$\begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}.$$

Calculation on the left side give us [8]:

$$\begin{bmatrix} u_{11} & u_{12} & u_{13} \\ l_{21}u_{11} & l_{21}u_{12} + u_{22} & l_{21}u_{13} + u_{23} \\ l_{31}u_{11} & l_{31}u_{12} + l_{32}u_{22} & l_{31}u_{13} + l_{32}u_{23} + u_{33} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}.$$

The main idea of the LU decomposition is to record the steps used in Gaussian elimination on A in the places where the zero is produced [22].

Example 2.1: Consider the matrix

$$A = \begin{bmatrix} 7 & 6 & 10 \\ 3 & 8 & 7 \\ 3 & 5 & 5 \end{bmatrix},$$

where $L = \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix}$ and $U = \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$.

Multiplication LU and setting the answer equal to A gives,

$$\begin{bmatrix} u_{11} & u_{12} & u_{13} \\ l_{21}u_{11} & l_{21}u_{12} + u_{22} & l_{21}u_{13} + u_{23} \\ l_{31}u_{11} & l_{31}u_{12} + l_{32}u_{22} & l_{31}u_{13} + l_{32}u_{23} + u_{33} \end{bmatrix} = \begin{bmatrix} 7 & 6 & 10 \\ 3 & 8 & 7 \\ 3 & 5 & 5 \end{bmatrix}.$$

Now we have to use this to find the entries in L and U. We begin by running along the top row to see that

$$u_{11} = 7; \;\; u_{12} = 6 \;\; ; u_{13} = 10 .$$

Now consider the second row

$$l_{21}u_{11} = 3 \implies l_{21} \times 7 = 3 \implies l_{21} = 0.4286\,;$$

$$l_{21}u_{12} + u_{22} = 8 \implies 0.4286 \times 6 + u_{22} = 8 \implies u_{22} = 5.4286\,;$$

$$l_{21}u_{13} + u_{23} = 7 \implies 0.4286 \times 10 + u_{23} = 7 \implies u_{23} = 2.7143\,.$$

At each step, the equation has only one unknown in it, and other quantities that we have already found. This pattern continues on the last row

$$l_{31}u_{11} = 3 \implies l_{31} \times 7 = 3 \implies l_{31} = 0.4286\,;$$

$$l_{31}u_{12} + l_{32}u_{22} = 8 \implies 0.4286 \times 6 + l_{32} \times 5.4286 = 5 \implies l_{32} = 0.4474\,;$$

$$l_{31}u_{13} + l_{32}u_{23} + u_{33} = 5 \implies 0.4286 \times 10 + 0.4474 \times 2.7143 + u_{33} = 5 \implies$$

$$\implies u_{33} = -0.5\,.$$

We have shown that

$$A = \begin{bmatrix} 7 & 6 & 10 \\ 3 & 8 & 7 \\ 3 & 5 & 5 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0.4286 & 1 & 0 \\ 0.4286 & 0.4474 & 15 \end{bmatrix} \begin{bmatrix} 7 & 6 & 10 \\ 0 & 5.4286 & 2.7143 \\ 0 & 0 & -0.5 \end{bmatrix}.$$

2.2. Eigenvalues and Eigenvectors

Let A be an $n \times n$ matrix. A scalar λ is an eigenvalue of A if there exists a non-zero vector V, such that [23]

$$AV = \lambda V.$$

In this case, vector V is called an eigenvector of A corresponding to λ.

Letting A be a $n \times n$ square matrix

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

with eigenvalue λ, then the corresponding eigenvectors satisfy [1, 10]

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} = \lambda \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}.$$

One can rewrite it in the form:

$$\begin{vmatrix} a_{11}-\lambda & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22}-\lambda & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn}-\lambda \end{vmatrix} \begin{vmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{vmatrix} = \begin{vmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{vmatrix},$$

or compactly :

$$(A-\lambda I)V = 0\,,$$

where I is the identity matrix. So the eigenvalue of A are computed as solutions of equation [23]:

$$det(A-\lambda I) = 0\,. \qquad (2.1)$$

This equation (2.1) is known as the characteristic equation of A.

Example 2.2 : Find eigenvalues and eigenvectors of a 2×2 Matrix.

$$A = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix}.$$

The characteristic equation is

$$|A-\lambda I| = \left|\begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix}\right| = 0\,,$$

$$\left|\begin{bmatrix} -\lambda & 1 \\ -2 & -3-\lambda \end{bmatrix}\right| = \lambda^2 + 3\lambda + 2 = 0\,.$$

Two eigenvalues are

$$\lambda_1 = -1; \lambda_2 = -2\,.$$

Let's find the eigenvector V_1, which is associated the eigenvalue, $\lambda_1 = -1$.

$$AV_1 = \lambda_1 V_1\,;$$

$$(A-\lambda_{1)} V_1 = 0\,;$$

$$\begin{bmatrix} -\lambda & 1 \\ -2 & -3-\lambda \end{bmatrix} V_1 = 0\,;$$

$$\begin{bmatrix} 1 & 1 \\ -2 & -2 \end{bmatrix} V_1 = \begin{bmatrix} 1 & 1 \\ -2 & -2 \end{bmatrix} \begin{bmatrix} v_{1,1} \\ v_{1,2} \end{bmatrix} = 0\,.$$

From the top row of the equations we get

$$v_{1,2} + v_{1,2} = 0,$$

$$v_{1,1} = -v_{1,2}.$$

Note that if we took the second row we would get

$$-2v_{1,1} + -2v_{1,2} = 0,$$

$$v_{1,1} = -v_{1,2}.$$

In any case we find that the first eigenvector is any two element column vector where the two elements have equal magnitude and opposite sign,

$$V_1 = k_1 \begin{bmatrix} +1 \\ -1 \end{bmatrix}.$$

Considering the same procedure for the second eigenvalue:

$$AV_2 = \lambda_2 V_2;$$

$$(A - \lambda_2)V_2 = \begin{bmatrix} -\lambda_2 & 1 \\ -2 & -3 - \lambda_2 \end{bmatrix};$$

$$\begin{bmatrix} 2 & 1 \\ -2 & -1 \end{bmatrix} v_2 = \begin{bmatrix} 2 & 1 \\ -2 & -1 \end{bmatrix} \begin{bmatrix} v_{2,1} \\ v_{2,2} \end{bmatrix} = 0;$$

$$2v_{2,1} + 1v_{2,2} = 0;$$

$$2v_{2,1} = -v_{2,2};$$

$$V_2 = k_2 \begin{bmatrix} +1 \\ -2 \end{bmatrix}.$$

Again, the choice of $+1$ and -2 for the eigenvector was random; only their ratio is important.

In Matlab this would work as follows:

```
>> A=[0 1;-2 -3]

A =

     0     1

    -2    -3

>> [v,lambda]=eig(A)

v =

    0.7071   -0.4472

   -0.7071    0.8944

lambda =

    -1     0

     0    -2
```

2.3. Spectral Decomposition

Let A be a regular $n \times n$ matrix, C be the $n\times n$ matrix formed by by the orthonormal eigenvectors of A.

An $n \times n$ matrix A is diagonalizable if there is an invertible $n \times n$ matrix C such that $C^{-1}AC$ is a diagonal matrix. The matrix C is said to diagonalize A. Here C^{-1} is inverse matrix of C [4].

Assume A has eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_n$ and corresponding linearly independent eigenvectors $V_1, V_2, \dots, V_n$ which can be denoted [24]

$$\begin{bmatrix} v_{11} \\ v_{12} \\ \vdots \\ v_{1n} \end{bmatrix}, \begin{bmatrix} v_{21} \\ v_{22} \\ \vdots \\ v_{2n} \end{bmatrix}, \dots, \begin{bmatrix} v_{n1} \\ v_{n2} \\ \vdots \\ v_{nn} \end{bmatrix}.$$

Define the matrices C and D composed of corresponding eigenvectors and eigenvalues:

$$C = \begin{bmatrix} v_{11} & v_{21} & \dots & v_{n1} \\ v_{12} & v_{22} & \dots & v_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ v_{1n} & v_{2n} & \dots & v_{nn} \end{bmatrix}; \ D = \begin{bmatrix} \boldsymbol{\lambda}_1 & 0 & \dots & 0 \\ 0 & \boldsymbol{\lambda}_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \boldsymbol{\lambda}_n \end{bmatrix}.$$

Here D is a diagonal matrix. Then [24].

$$AC = A[V_1 \, V_2 \ \dots \ V_n] = [AV_1 \, AV_2 \ \dots \ AV_n] = [\lambda_1 V_1 \, \lambda_2 V_2 \ \dots \ \lambda_n V_n] =$$

$$= \begin{bmatrix} \lambda_1 v_{11} & \lambda_2 v_{21} & \dots & \lambda_1 v_{n1} \\ \lambda_1 v_{12} & \lambda_2 v_{22} & \dots & \lambda_2 v_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_1 v_{1n} & \lambda_2 v_{2n} & \dots & \lambda_n v_{nn} \end{bmatrix} = \begin{bmatrix} v_{11} & v_{21} & \dots & v_{n1} \\ v_{12} & v_{22} & \dots & v_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ v_{1n} & v_{2n} & \dots & v_{nn} \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{bmatrix} = CD.$$

Next formula gives decomposition of A with a similarity transformation involving C and D [24].

$$A = CDC^{-1}.$$

Example 2.2 : Find the matrix that diagonalizes

$$A = \begin{bmatrix} 1 & 2 & 0 \\ 0 & 3 & 0 \\ 2 & -4 & 2 \end{bmatrix}.$$

First we'll find the eigenvalues and eigenvectors of A. This matrix has 3 eigenvalues

$$\lambda_1 = 3, \lambda_2 = 2, \ \lambda_3 = 1.$$

A is a 3×3 matrix with three different eigenvalues; therefore, it is diagonalizable. Note that if there are exactly n different eigenvalues in an $n \times n$ matrix then this matrix is diagonalizable.

The eigenvectors of A are

$$v_1 = \begin{bmatrix} -1 \\ -1 \\ 2 \end{bmatrix}, v_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, v_3 = \begin{bmatrix} -1 \\ 0 \\ 2 \end{bmatrix}.$$

Now, let C be the matrix with these eigenvectors as its columns:

$$\begin{bmatrix} -1 & 0 & -1 \\ -1 & 0 & 0 \\ 2 & 1 & 2 \end{bmatrix}.$$

Then C diagonalizes A, as a simple computation confirms, having calculated C^{-1} using any suitable method:

$$C^{-1}AC = \begin{bmatrix} 0 & -1 & 0 \\ 2 & 0 & 1 \\ -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 2 & 0 \\ 0 & 3 & 0 \\ 2 & -4 & 2 \end{bmatrix} \begin{bmatrix} -1 & 0 & -1 \\ -1 & 0 & 0 \\ 2 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

2.4. Singular Decomposition

The singular value decomposition of a matrix A is the factorization of A into the product of three matrices $A = U\sum V^T$ where the columns of U and V are orthonormal and the matrix $\sum$ is diagonal with positive real entries [12].

Singular value decomposition takes a rectangular matrix A, where A is a $n \times k$ matrix. The SVD theorem states:

$$A_{nxk} = U_{nxn} \Sigma_{nxk} V_{kxk}^T ,$$

where,

$$U^T U = I_{nxn} ;$$

$$V^T V = I_{kxk} .$$

Calculating the SVD consists of finding the eigenvalues and eigenvectors of AA^T and $A^T A$. The eigenvectors of $A^T A$ make up the columns of V , the eigenvectors of AA^T make up the columns of U. Also, the singular values in $\sum$ are square roots of eigenvalues from AA^T or $A^T A$. The singular values are the diagonal entries of the $\sum$ matrix and are arranged in descending order. The singular values are always real numbers. If the matrix A is a real matrix, then U and V are also real. Let λ_i is eigenvalues of $W = AA^T$ and $L = A^T A$ matrixes and let u_i and v_i are corresponding eigenvectors of W and L. Let's construct three matrices from these values: the diagonal matrix $\sum$, which has $\sigma_i =$

$\sqrt{\lambda_i}$ values on the diagonal (padded with zeros if we run out of σ_s); the matrix U with u_i as columns; and the matrix V with v_i as the columns. (As an example, consider an A that is 2×4; then, U will be 4x4, $\sum$ will be 4×2, with the rightmost 2 columns being just zeros, and V will be 2×2) [2, 5].

Example 2.3 : Let's look at the example of 4×2 matrix A ,and perform $U\sum V^T$ multiplication step-by-step in matlab [20].

$$A = \begin{bmatrix} 2 & 4 \\ 1 & 3 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

Let's construct W matrix and find the eigenvalues and eigenvectors.

$$W = AA^T = \begin{bmatrix} 2 & 4 \\ 1 & 3 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 2 & 1 & 0 & 0 \\ 4 & 3 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 20 & 14 & 0 & 0 \\ 14 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

```
>> A=[2 4;1 3;0 0;0 0]

A =

     2     4

     1     3

     0     0

     0     0

>> W=A*transpose(A)

W =

    20    14     0     0

    14    10     0     0

     0     0     0     0

     0     0     0     0
```

Now that we have a 4×4 matrix we can determine the eigenvalues and eigenvectors of the matrix W.

$$\begin{bmatrix} 20-\lambda & 14 & 0 & 0 \\ 14 & 10-\lambda & 0 & 0 \\ 0 & 0 & -\lambda & 0 \\ 0 & 0 & 0 & -\lambda \end{bmatrix} U = (W - \lambda I)U = 0.$$

Thus from the solution of the characteristic equation $|W - \lambda I| = 0$ we obtain: $\lambda_1 = 0.1339$, $\lambda_2 = 29.8661$, $\lambda_3 = 0$, $\lambda_4 = 0$. This value can be used to determine the eigenvector that can be placed in the columns of U.

Combining these we obtain we can construct matrix U.

$$U = \begin{bmatrix} -0.8174 & -0.5760 & 0 & 0 \\ -0.5760 & 0.8174 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Now let's construct matrix L and determine the eigenvalues and eigenvectors.

$$L = A^T A = \begin{bmatrix} 2 & 1 & 0 & 0 \\ 4 & 3 & 0 & 0 \end{bmatrix} \begin{bmatrix} 2 & 4 \\ 1 & 3 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 5 & 11 \\ 11 & 25 \end{bmatrix}.$$

A similarly, with eigenvectors of 2×2 matrix L we can construct 2×2 matrix V matrix.

$$V = \begin{bmatrix} -0.4046 & -0.9145 \\ -0.9145 & 0.4046 \end{bmatrix}.$$

Finally by the square root of the eigenvalues W and V we can construct 4×2 matrix $\sum$.

$$\sum = \begin{bmatrix} 5.4630 & 0 \\ 0 & 0.3660 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

All these processes we can make in matlab and we can get singular the composition of given A matrix by using simple svd(A) function [20].

```
>> A=[2 4;1 3;0 0;0 0]

A =

     2     4

     1     3

     0     0

     0     0

>> [U,S,V]=svd(A)

U =

   -0.8174   -0.5760         0         0

   -0.5760    0.8174         0         0

         0         0    1.0000         0

         0         0         0    1.0000

S =

    5.4650         0

         0    0.3660

         0         0

         0         0

V =

   -0.4046   -0.9145

   -0.9145    0.4046
```

3. SINGULAR SPECTRUM ANALYSIS IN THE STUDY OF TIME SERIES

Singular spectrum analysis (SSA) is a technique of time series analysis and forecasting combining elements of classical time series analysis, multivariate statistics, multivariate geometry, dynamical systems and signal processing. SSA seeks to decompose the original series into a sum of a small number of interpretable components such as trend, oscillatory components and noise. It is based on the singular value decomposition of a specific matrix constructed upon the time series [17].

The basic SSA primarily involves two stages: decomposition and reconstruction. The decomposition consists of embedding and singular value decomposition (SVD). The reconstruction stage consists of eigentriple grouping and diagonal average. These two stages make up the basic SSA algorithm [16].

3.1. Time series

Time Series is an ordered sequence of values of a variable at equally spaced time intervals. Data that can be classified as time series include annual rainfall, daily or weekly closing price of stock, number of death cases in the year and recording of temperature. A time series that can be measured as a single variable is termed as univariate. If two or more variables are measured then we call it multivariate. When time series is measured at discrete or finite steps or points, then it is a *discrete time series*. The data set used in this work is a discrete time series. The mathematical expression for a discrete time series is x_t; $t = 0,1,2, \dots$. In effect, x_t is considered to be a random variable. However, observations which are measured over a specified interval is known as a continuous time series [18].

Generally speaking, a time series has four major components, namely: *seasonal, cyclical, trend and irregular* . Seasonal variation in time series occurs as a result of changes in the weather and climate conditions. For example, the increase in the sales of winter clothes is caused by seasonal variation. Repeated patterns or cycles due to medium term changes as seen in the financial markets lead to cyclical variation. Eventualities which are not repetitive in nature like earthquakes, flood, war and other natural disasters create a scenario in time series referred to as irregular or random fluctuations. A trend in time series occurs when there is a pattern of continuous increase, decrease or stagnation over time. For example, there is an upward trend in the rent of city apartments and a downward trend in birth rates [16, 18]. Time series in Figure 3.1 is trend.

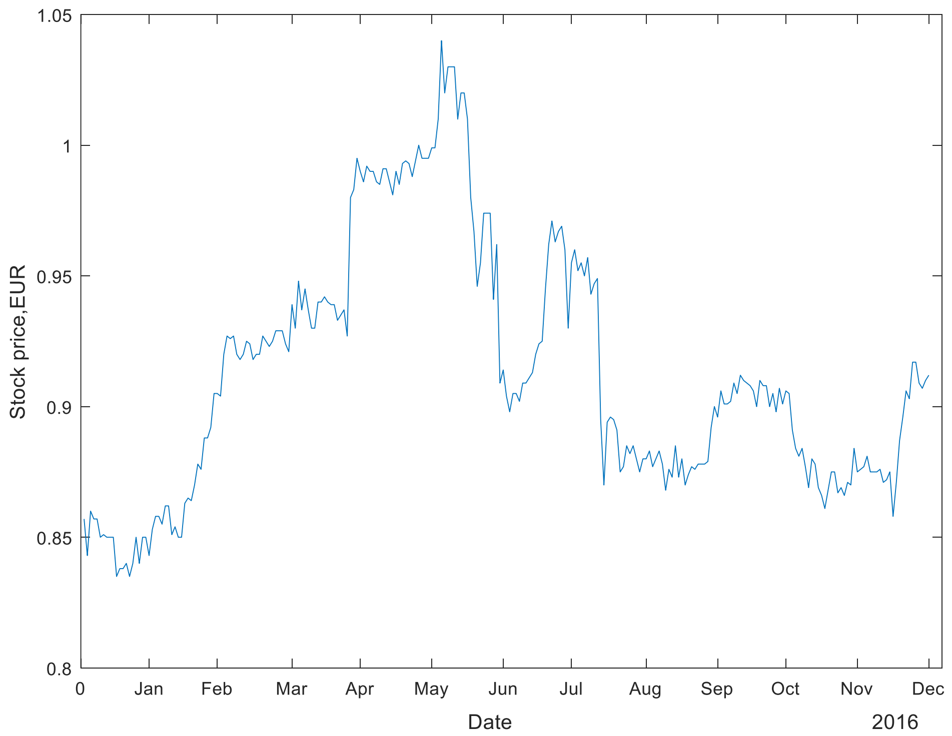


Figure 3.1. The time series for AS Tallink Grupp stock prices in 2016 year.

The usage of time series models is twofold:

- Obtain an understanding of the underlying forces and structure that produced the observed data;
- Fit a model and proceed to forecasting, monitoring or even feedback and feedforward control.

Time Series Analysis is used for many applications such as: Economic Forecasting, Sales Forecasting, Budgetary Analysis, Stock Market Analysis, Census Analysis, Yield Projections, Process and Quality Control, Inventory Studies, Utility Studies, Workload Projections, and etc. [11].

One of the main problems related time series is time series forecasting. Time series forecasting uses information regarding historical values and associated patterns to predict future activity. Most often, this refers to trend analysis, cyclical fluctuation analysis and issues of seasonality [21].

3.2. Singular Value Decomposition of Time Series

1st step: Embedding

The purpose of the first step is mapping the original time series into the trajectory matrix. Consider $X = (x_1, \dots, x_N)$, the time series of length N, where N is greater than 2 and X is a nonzero series; that is there exists at least one i such that $x_i \neq 0$. Let L, be some integer called the *window length*, which is $1 < L < N$. Then let $K = N - L + 1$. To perform the embedding we map the initial time series into a sequence of lagged vectors of size L by forming $K = N - L + 1$ lagged vectors

$$X_i = (x_i, \dots, x_{i+L-1})^T \ (1 \leq i \leq K)$$

of size L. The trajectory matrix of the series X is

$$\mathbf{X} = [X_1, \dots, X_N] = (x_{ij})_{i,j=1}^{L,K} = \begin{pmatrix} x_1 & x_2 & x_3 & \cdots & x_K \\ x_2 & x_3 & x_4 & \cdots & x_{K+1} \\ x_3 & x_4 & x_5 & \cdots & x_{K+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_L & x_{L+1} & x_{L+2} & \vdots & x_N \end{pmatrix}.$$

The rows and columns of trajectory matrix **X** are subseries of the original series. The matrix X is *Hankel matrix* which mean that the (i, j)th component of the matrix is **X** is $x_{ij} = x_{i+j-1}$ and **X** takes the same value for a constant value of $i + j$ =const [17].

2nd step: Singular Value Decomposition (SVD)

SVD is applied to the trajectory matrix **X** at this step. Let $S = \mathbf{X}\mathbf{X}^{\mathbf{T}}$ and denote by $\lambda_1, \lambda_2, \dots, \lambda_L$ the eigenvalues of S in decreasing order, $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_L \geq 0$. Let $U_1, U_2, \dots, U_L$ be the orthonormal eigenvectors of the matrix S corresponding to those eigenvalues.

Let $V_i = \frac{\mathbf{X}^{\mathbf{T}} U_i}{\sqrt{\lambda_i}} \, (i = 1,2, \dots, d)$, where d equal to the rank of the matrix **X** is the maximum of i such $\lambda_i > 0$. In real-life series usually $d = \min\{L, K\}$.

The triple $(\sqrt{\lambda_i}, V_i, U_i)$ is called as ith $eigentriple$ of the SVD [17].

3.3. Reconstruction

1st step: Eigentriple Grouping

The eigentriple grouping step corresponds to splitting the elementary matrices $\mathbf{X_i}$ into several groups and summing the matrices in each group. Let $I = \{I_1, \dots, I_m\}$, where each I_j contains several $\mathbf{X_i}$'s, where $\mathbf{X_i} = \sqrt{\lambda_i} V_i^T U_i$.

$$\widetilde{\mathbf{X}} = \mathbf{X}_{I_1} + \mathbf{X}_{I_2} + \cdots + \mathbf{X}_{I_m} .$$

The procedure of the set $I_1, \dots, I_m$ is called the eigentriple grouping. If $m = d$ with $I_j = \{j\},\ j = 1, \dots, d$, then the procedure is called elementary grouping [17].

2nd step: Diagonal Averaging

At diagonal averaging step each matrix $\mathbf{X}_{I_j}$ is going to be transformed into a new series with length N . Let Y be an LxK matrix y_{ij} is element of **Y. Y** can be transferred to series $y_1, y_2, \dots, y_N$ by

$$y_k = \begin{cases} \dfrac{1}{k}\displaystyle\sum_{m=1}^{k} y^*_{m,k-m+1} & for\ 1 \le k < L^*; \\ \dfrac{1}{L^*}\displaystyle\sum_{m=1}^{L^*} y^*_{m,k-m+1} & for\ L^* \le k \le K^*; \\ \dfrac{1}{N-k+1}\displaystyle\sum_{m=k-K^*+1}^{N-K^*+1} y^*_{m,k-m+1} & for\ K^* < k \le N . \end{cases}$$

Here $1 \le i \le L$, $1 \le j \le K$ and $L^* = \min(L, K), K^* = \max(L, K), N = L + K - 1, i + j = k + 1$. For example, the choice $k = 1$ gives $y_1 = y_{1,1}$; for $k = 2$ we have $y_2 = (y_{1,2} + y_{2,1})/2$; for $k = 3$ $y_3 = (y_{1,3} + y_{3,1} + y_{2,2})/3$.

Diagonal averaging applied to a resultant matrix $\mathbf{X}_{I_k}$, it produces $\tilde{X}^{(k)} = (\tilde{x}_1^{(k)}, \tilde{x}_2^{(k)}, \dots, \tilde{x}_N^{(k)})$, where $\tilde{X}^{(k)}$ is reconstructed series. The original series X is decomposed into the sum of reconstructed series [17]

$$x_n = \sum_{k=1}^{m} \tilde{x}_n^{(k)}\ (n = 1,2, \dots N).$$

This decomposition is the main result of the SSA algorithm [17].

3.4. Forecasting Algorithm

Let us describe the SSA forecasting algorithm [6, 17, 25]:

- Consider the initial time series $X = (x_1, \dots, x_N)$ with length N, where $N > 2$.
- Fix the window length L, $1 < L < N$.
- Construct the trajectory matrix $\mathbf{X} = [X_1, \dots, X_K]$ of the time series X.
- Construct orthonormal system of eigenvectors $U_1, \dots, U_L$ form the SVD of **X.**
- Estimate matrix $\widehat{\mathbf{X}} = \left[\hat{X}_1 : \dots : \hat{X}_K\right] = \sum_{i=1}^{L} U_i U_i^T \mathbf{X}$.
- Construct matrix $\widetilde{\boldsymbol{X}} = \mathcal{H}\widehat{\mathbf{X}}$=$[\tilde{X}_1, \dots, \tilde{X}_K]$. Here $\widetilde{\boldsymbol{X}}$ is the result of the Hankellization of the matrix $\widehat{\mathbf{X}}$.
- Set $v^2 = \pi_1^2 + \cdots + \pi_L^2$ where π_i is the last component of the vector U_i (i = 1, ..., L). It comes out that $v^2 < 1$.
- Determine vector $A = (\alpha_1, \dots, \alpha_{L-1})$:

$$A = \frac{1}{1 - v^2} \sum_{i=1}^{L} \pi_i \underline{U_i}.$$

It can be proved that in the nptations above, the last component x_L of any vector $X = (x_1, \dots, x_L)^T$ is a linear combination of the first components $(x_1, \dots, x_{L-1})$:

$$x_L = \alpha_1 x_{L-1} + \cdots + \alpha_{L-1} x_1.$$

- The last step is forecasting procedure. Define the time series $X_{N+h} = (x_1, \dots, x_{N+h})$ by the formulas.

$$x_i = \begin{cases} \tilde{x}_i & for\ i = 1, \dots, N\ ; \\ \sum_{j=1}^{L-1} \alpha_j x_{i-j} & for\ \ i = N+1, \dots, N+h. \end{cases}$$

Here $\tilde{x}_i$ $(i = 1, \dots, N)$ are the components of reconstructed series. The numbers $x_{N+1}, \dots, x_{N+h}$ form the h terms of the SSA recurrent forecast.

3.5. Choice of SSA parameters

The choice of parameters depends on the data we have and the analysis we have to perform [16]. We discuss the selection of SSA parameters separately for all the main problems of time series analysis.

There are two parameters in Basic SSA: the first is the window length L and the second parameter is the number of components r for reconstruction [9]. Values for L and r could be defined using information provided by the time series under study or through additional indices:

Selection of the window length L: The window length L is the only parameter in the decomposition stage. Selection of the proper window length depends on the problem in hand and on preliminarily information about the time series [7]. Knowing that the time series may have a periodic component with an integer period, to get a better separability of this components it is recommended to choose the window length proportional to that period. Theoretical results show that L should be large enough but not greater than $N/2$ [6].

The number of components r: The theory of separability, that is how well the components can be separated, is the basis for the definition of r. A main criterion is based on the contribution of each component to the variance of X, evaluated as λ_i/Γ $(\Gamma = \sum_{i=1}^{d} \lambda_i)$. Select r out of the components so that the sum of their contributions is at least a predetermined threshold, for example $\geq$ 90 % [9].

3.6. Forecast accuracy

There are three main approaches to evaluate the accuracy and reliability of forecasts in time series [19].

- construction of confidence intervals;
- assessment of retrospective forecasts;
- checking the stability of forecasts.

Although the combination of the mentioned three approaches are used in practice, in this section the second approach (assessment of retrospective forecasts) is researched.

Retrospective forecasts are usually performed by truncating the time series and by obtaining forecasts for points temporarily removed. These forecasts can then be compared with the observed values of the time series to assess their quality and reliability. Let $e_{T,h}(x) = y_{T+h}(x) - \hat{y}_{T,h}(x)$ denote the forecast error. Here $\hat{y}_{T,h}(x)$ are the forecast for $y_{T+h}(x)$. Then, a measure of accuracy define by the following formula [19]

$$ISE_{T,h} = \sum_{x} e_{T,h}^2(x).$$

4. NUMERICAL EXPERIMENTS: DATA AND RESULTS

This chapter involves the numerical experiments on the SSA algorithm for a given data. The idea of the experiment is to do the calculations step by step, using a short time series. The experiment relies on elementary linear algebra tools that given in section 1: a basic understanding of concepts such as matrix-vector products, eigenvalues and eigenvectors. Experiment is applied to the close prices of “AS Tallink Grupp” stock taken from Yahoo! Finance for period from January 1, 2016 to December 30, 2016 (See Appendix D). Experiment consist of four main parts: calculation of eigenvalues and eigenvectors; construction of the principal components of the time series; reconstruction and forecasting; results of the experiment. For experiment we use the software Matlab. Codes see in Appendix A, B, C.

4.1. Calculation of eigenvalues and eigenvectors

Consider the time series taken as a close prices of “AS Tallink Grupp” stock, stored in the vector X where X is called initial data. Initial data consist of $N = 261$ data points which means that length of the initial data equal 261. The graph of the time series given in Figure 4.1.

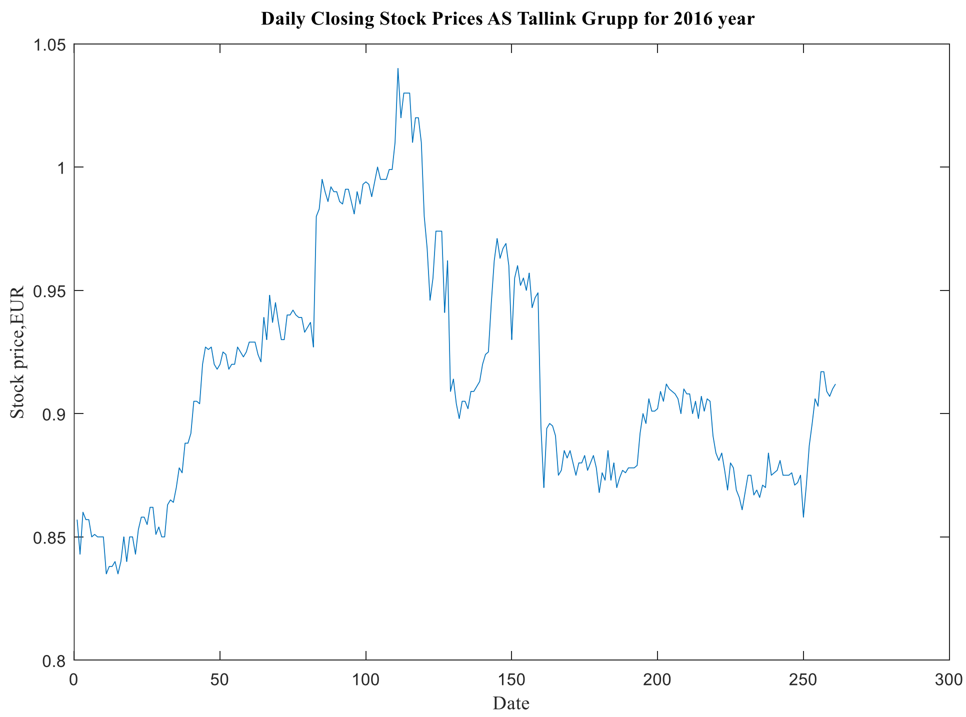
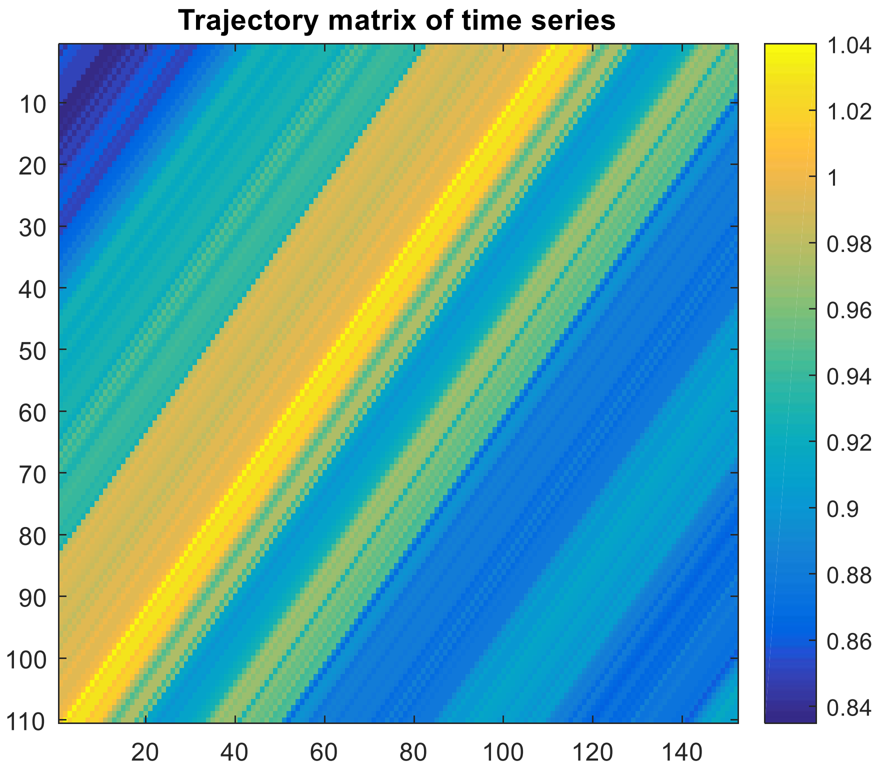


Figure 4.1. The time series of AS Tallink Grupp stock prices in 2016 year.

First need to calculate the $L \times L$ matrix S of $X(t)$ $(t = 1, \dots, N)$. This matrix can computed by creating the $L \times (N - L + 1)$ "trajectory matrix" that is formed by L lag-shifted copies of $X(t)$, which are $N - L + 1$ long. Figure 5.2 shows trajectory matrix of the time series $X(t)$.



$Figure$ 4.2. The trajectory matrix of the time series $X(t)$.

So refer section 3.5 the matrix S is computed for $L = 110$. Figure 4.3 and Figure 4.4 reveal corresponding eigenvalues and first eight eigenvectors of the matrix S.

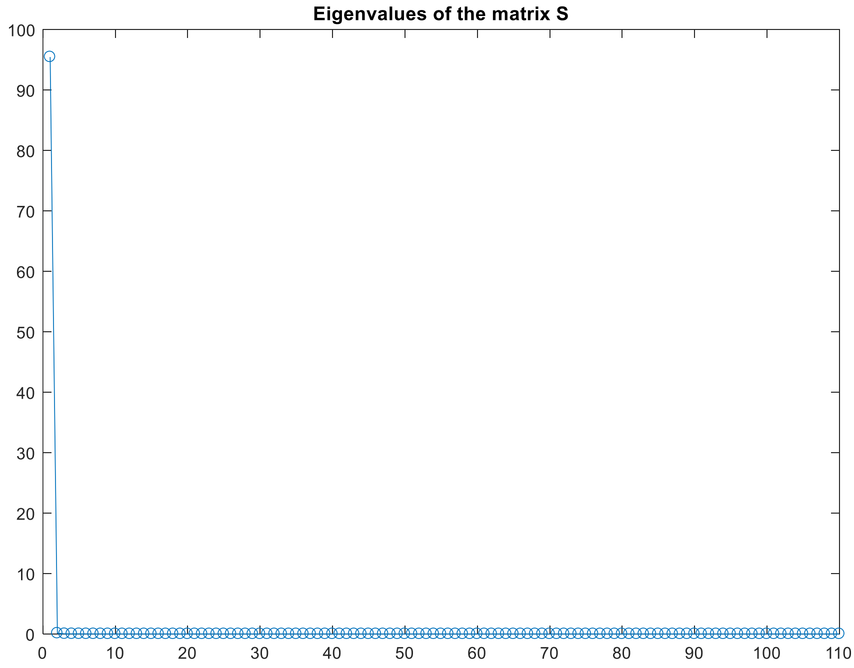


Figure 4.3. Eigenvalues of the matrix S.

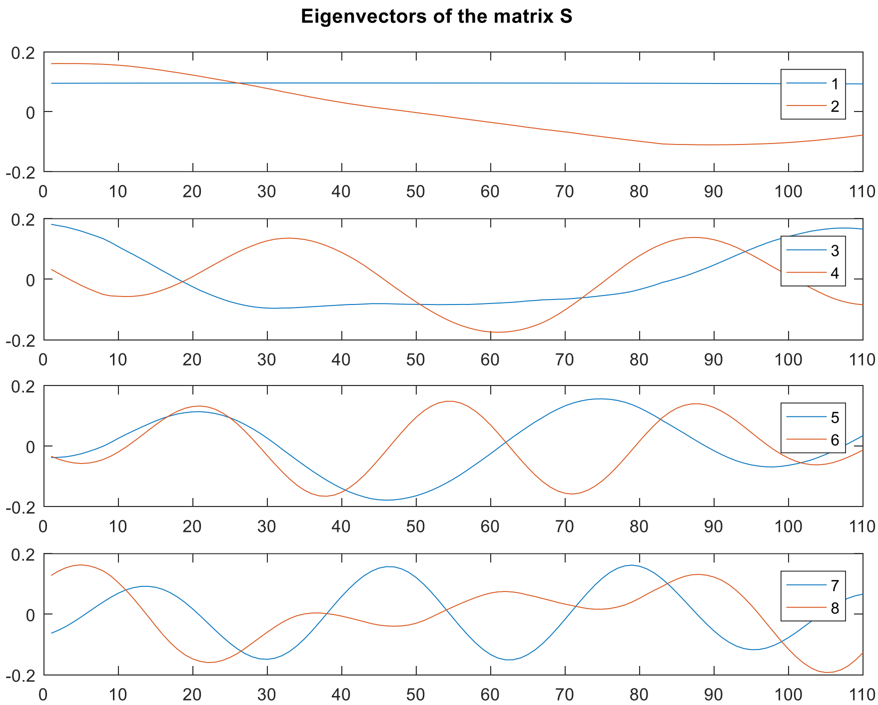


Figure 4.4. Eigenvectors of the matrix S.

4.2. Principal components

In theory of SSA are important also so called principal components of the time series $X(t)$.

An oscillation is characterized by a pair of almost equal SSA eigenvalues and related PCs that are in approximate phase quadrature. Such a pair can represent a nonlinear, a harmonic fluctuation, the principal components are again time series, of the same length as the trajectory matrix [15].

In Matlab, the principal components are computed by linear combination of the trajectory matrix $\mathbf{X}$ and the matrix of eigenvectors V (each column is one eigenvector) [3] what yields a matrix of size $(N - L + 1) \times L$.

$$PC = \mathbf{X}' \times V.$$

The columns of matrix PC are principal components of the initial time series. To refer this formula we can say that "the trajectory matrix is projected onto the eigenvectors". Figure 4.5 and Figure 4.6 show first and last four Principal Components of the time series.

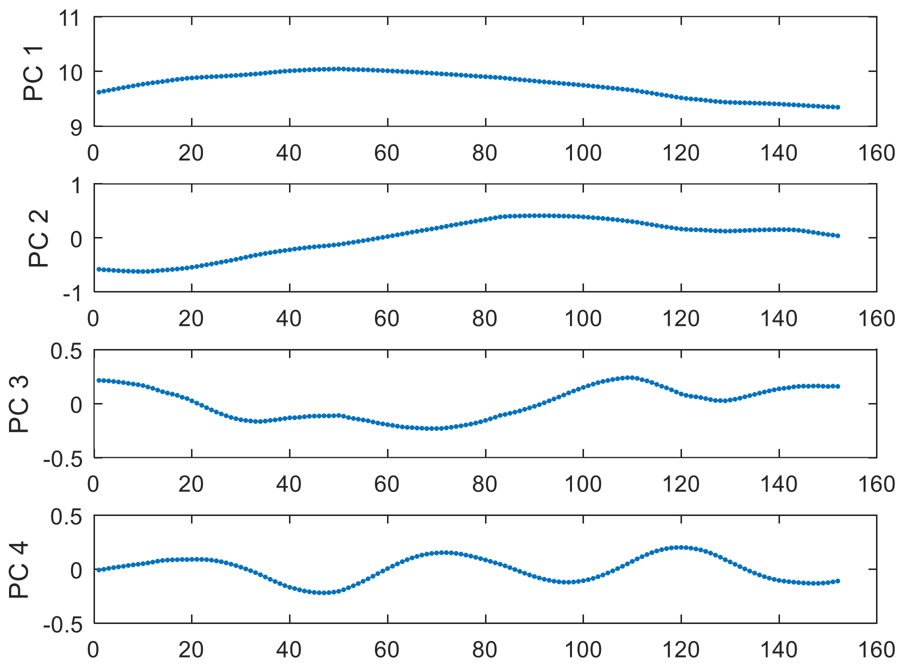


Figure 4.5. Principal Components of the time series.

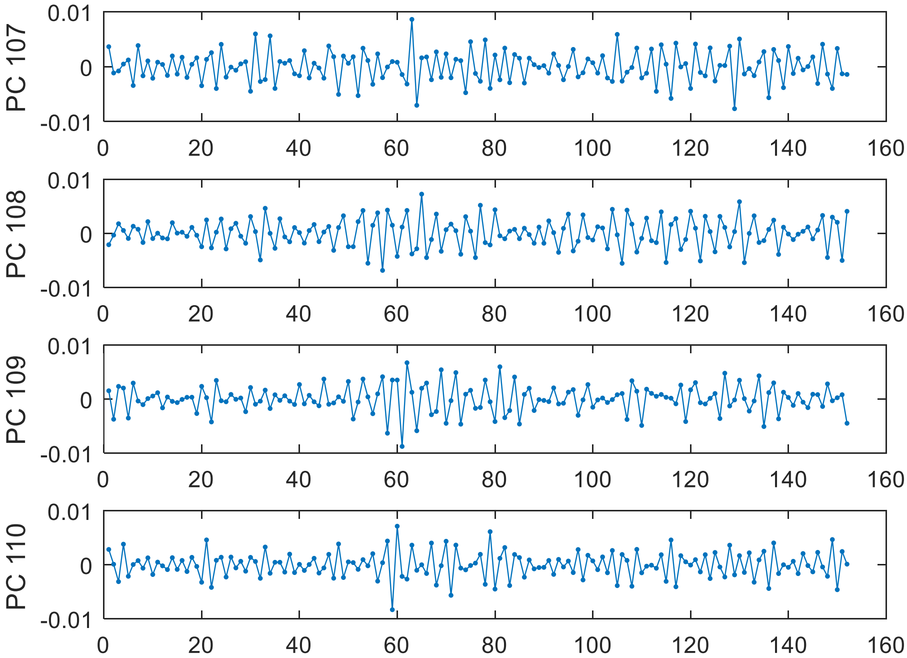


Figure 4.6. Principal Components of the time series .

4.3. Reconstruction of the time series

In order to determine the reconstructed components first we need to create $(N - L + 1) \times L$ matrix by invert projection of the PC and the matrix of eigenvectors V. Then averaging along anti-diagonals of this matrix gives the reconstructed components for the initial time series. The reconstructed components contain a matrix size of $(N \times L)$. Figure 4.7 and Figure 4.8 illustrate first and last four reconstructed components of the time series.

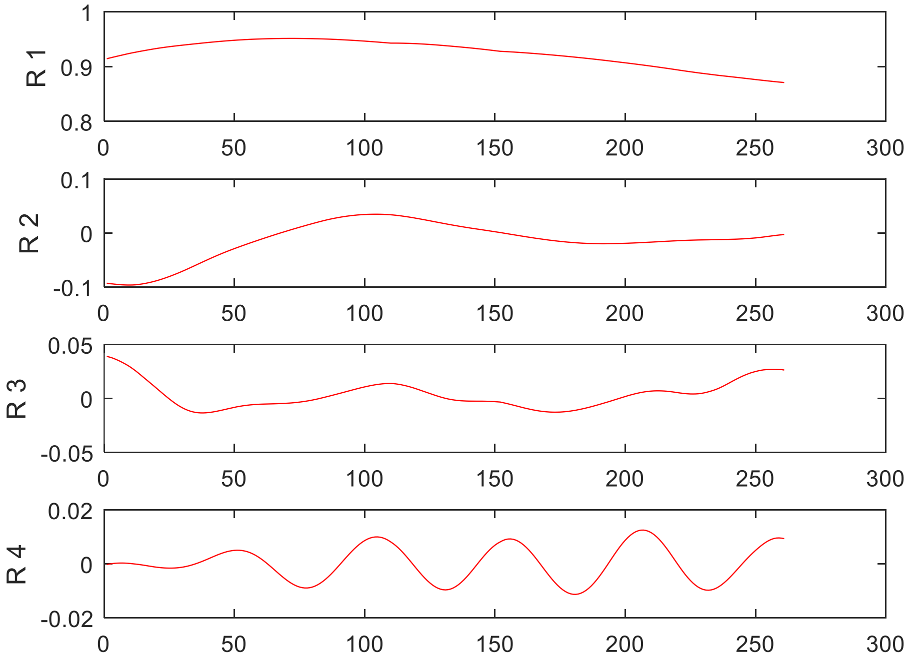


Figure 4.7. The reconstructed components of the time series.

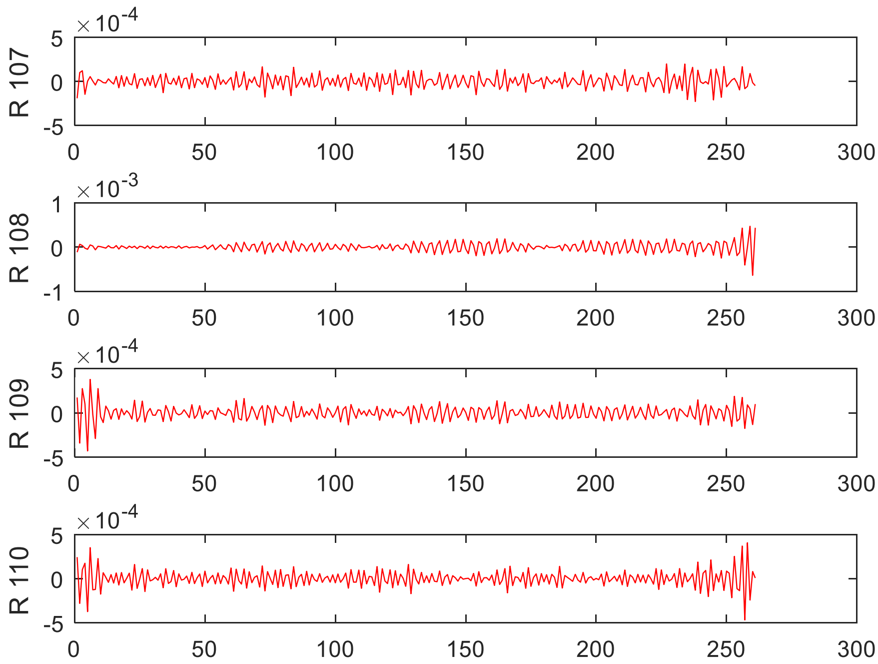


Figure 4.8. The reconstructed components of the time series.

According to the Figure 4.7 the first two reconstructed components contain practically all trends of the time series.

The following Figure 4.9 shows a comparison of initial time series and sum of N reconstructed components and also a comparison of initial time series and the sum of the first two reconstructed components. It is noticeable that the sum of N reconstructed components give initial time series.

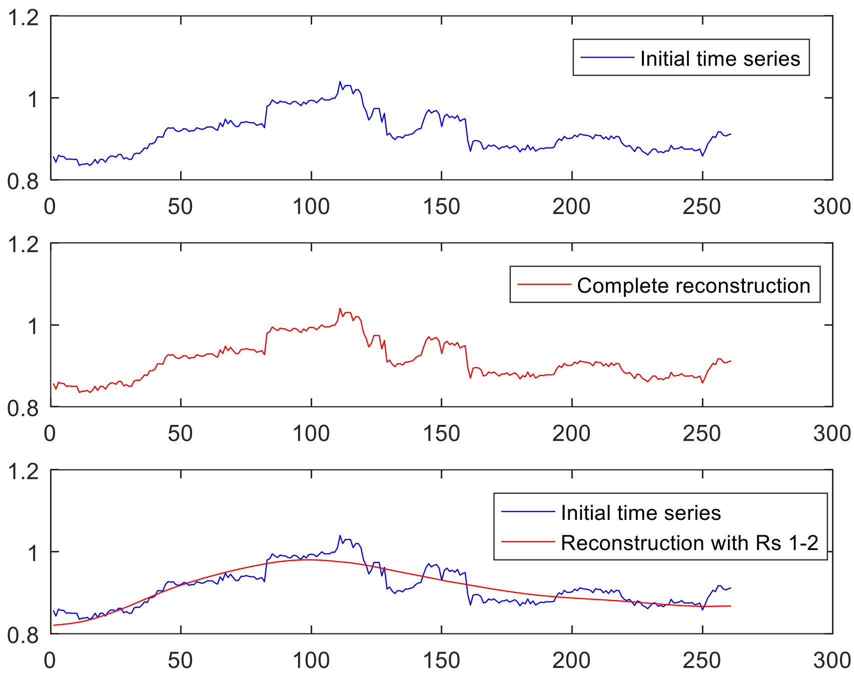


Figure 4.9. Comparison initial time series and reconstructed components.

4.4. Forecasting Method

In section 4.1-4.3 we applied main algorithm of SSA to the financial time series- the close prices of "AS Tallink Grupp" stock taken from Yahoo! Finance. First, we created trajectory matrix and found eigenvalues and eigenvectors. Then constructed principal and reconstructed components.

In this section the goal is to predict the next days close price by applying forecasting algorithm given in section 3.4. We continue by developing forecasts for the initial time series of "AS Tallink Grupp" stock prices in 2016 year shown in Figure 4.1. We begin by developing forecasting for points $N+1, N+2, \dots, N+h$. First of all we determine h. In this study we predict next 33 days. So we take $h = 33$ points. Then vector $A = (\alpha_1, \dots, \alpha_{L-1})$ is determined by given formula in section 3.4:

$$A = \frac{1}{1-v^2}\sum_{i=1}^{r} \pi_i \underline{U_i}.$$

Here v^2 is calculated by sum squares of the last component of the eigenvector (section 3.4). In the last step define the time series $X_{N+h} = (x_1, \ldots, x_{N+h})$ by the formula (section 3.4)

$$x_i = \begin{cases} \tilde{x}_i & for\ i = 1, \ldots, N\ ; \\ \sum_{j=1}^{L-1} \alpha_j x_{i-j} & for\ \ i = N+1, \ldots, N+h. \end{cases}$$

The following Figure 4.10 compares the original time series where last 33 points are predicted by SSA forecasting algorithm and complete reconstructed components for $(N + 33)$ points where last 33 points calculated by steps given in section 3.4.

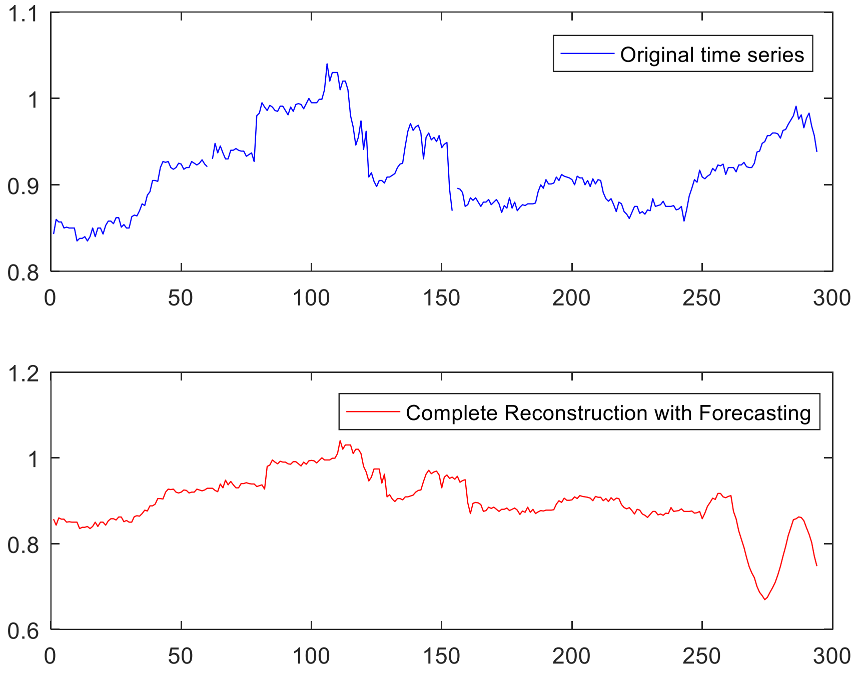


Figure 4.10. Comparison of original time series and complete reconstruction with forecasting.

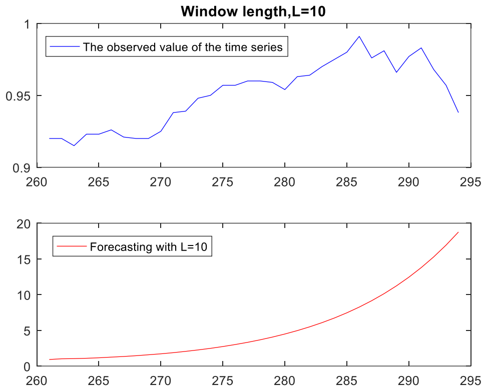
4.5. Interpretation of results

.

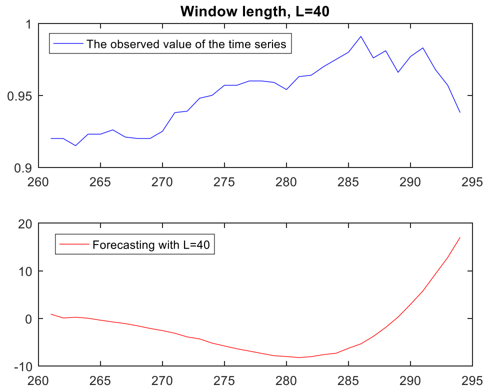
Before numerical experiments on the SSA algorithm, the window length and number of components r need to be decided. Our next task in this research is to find a suitable sampling for window length L and number of components r.

Influence of the window length L on forecasting. According to section 3.5, window length should be large enough but not greater than $N/2$. So in experiments in section 4.1-4.4 as window length we chose $L = 110$ and got forecasting shown in Figure 4.10. Figure 4.11

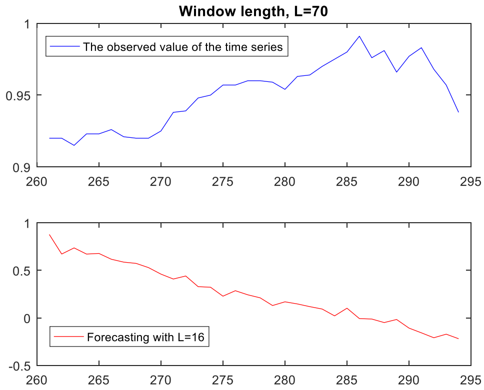
illustrate comparison of the observed value of the time series and forecast for next $(N + 33)$ points. The task is conducted for different window length $(L = 10.40.70, 110)$.



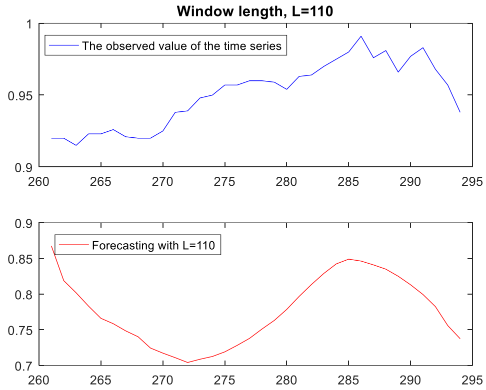
(a) $L = 10$



(b) $L = 40$



(c) $L = 70$



(d) $L = 110$

Figure 4.11. Comparison of the observed value of the time series and forecast.

For each forecast we can measure accuracy by the formula given in section 3.6:

$$ISE_{T,h} = \sum_x e_{T,h}^2 (x).$$

$e_{T,h}(x)$ is forecast error and calculated by next formula:

$$e_{T,h}(x) = y_{T+h}(x) - \hat{y}_{T,h}(x).$$

So, from calculation of forecast accuracy we get next results:

$$ISE_{T,h,1} = 1570 \text{ , when } L = 10;$$

$$ISE_{T,h,2} = 1456{,}8 \text{ , when } L = 40;$$

$$ISE_{T,h,3} = 19{,}8920, \text{ when } L = 70;$$

$$ISE_{T,h,3} = 1{,}0936 \text{ , when } L = 110.$$

From calculations it is clear that,

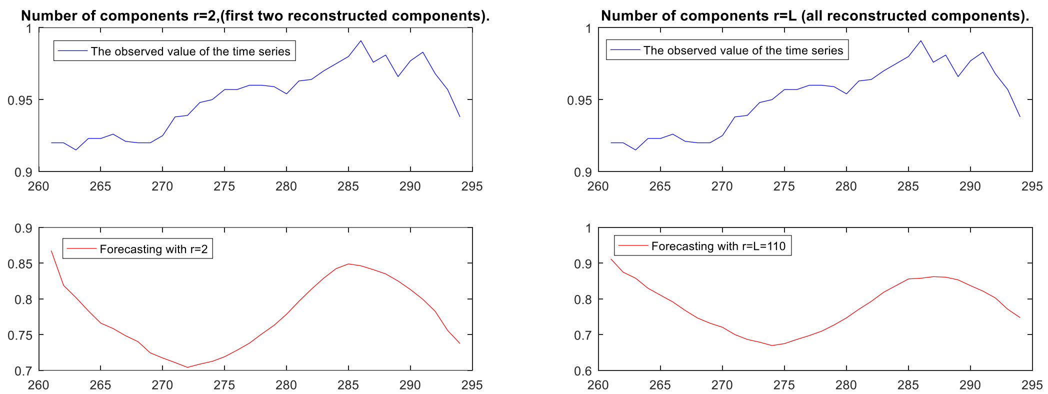
$$\max(ISE_{T,h,1}, ISE_{T,h,2}, ISE_{T,h,3}, ISE_{T,h,3} = \ 1570$$

and

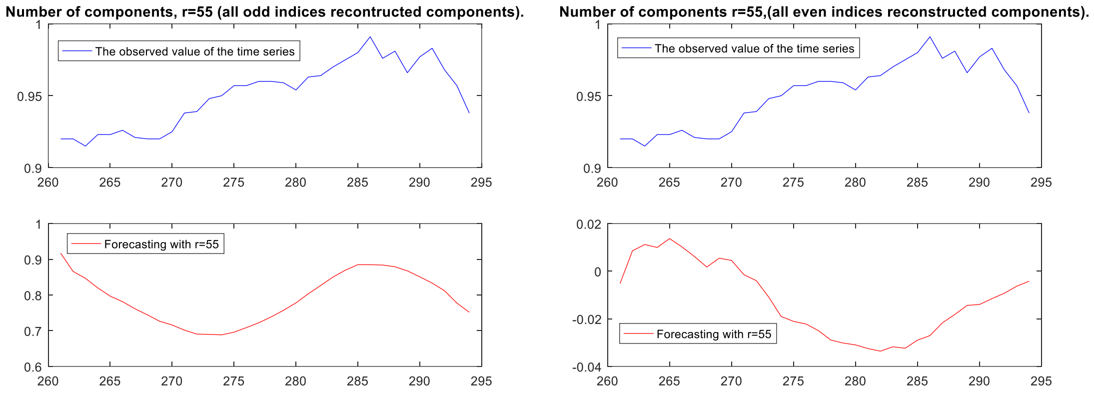
$$\min\,(ISE_{T,h,1}, ISE_{T,h,2}, ISE_{T,h,3}, ISE_{T,h,3} = 1{,}0936.$$

It mean when L is increased the forecast accuracy gets smaller and the forecast is more exact. Therefore, it would be better to work with window length $L = 110$.

Influence of the number of components r on forecasting. For preferable results, the choice of s r should be made accordingly to initial data and intended experiments. In the literatures several ways shown to determine r. One of them is based on the contribution of each component to the variance of X, evaluated as λ_i/Γ $(\Gamma = \sum_{i=1}^{d} \lambda_i)$. In experiment which is conducted in section 4.4, we used the sum of all reconstruction components and got result shown in Figure 4.10. Figure 4.12 illustrate comparison of the observed value of the time series and forecast for next $(N + 33)$ points with different selection of r.



(a) (b)



(c) (d)

Figure 4.12. Comparison of the observed value of the time series and forecast.

(a) $r = 2$, *first two reconstructed components*;

(b) $r = L = 110$, *all reconstructed components*;

(c) $r = 55$, *all odd indices reconstructed components*;

(d) $r = 55$, *all even reconstructed components*.

$ISE_{T,h,1} = 1{,}0936$, when $r = 2$ (sum of first two reconstructed components);

$ISE_{T,h,2} = 1{,}0005$, when $r = 55$ (sum of odd indices reconstructed components);

$ISE_{T,h,3} = 31{,}5650$, when $r = 55$ (sum of even indices reconstructed components);

$ISE_{T,h,3} = 1{,}1533$, when $r = L = 110$ (sum of all reconstructed components).

From calculations it is clear that,

$$\max(ISE_{T,h,1}, ISE_{T,h,2}, ISE_{T,h,3}, ISE_{T,h,3} = 31{,}5650$$

and

$$\min\,(ISE_{T,h,1}, ISE_{T,h,2}, ISE_{T,h,3}, ISE_{T,h,3} = 1{,}0005 \ .$$

The calculations above show that by using even indices reconstructed components we get high forecast accuracy. All others case forecast accuracy doesn’t change significantly. Comparison of some results see in Appendix E.

5. CONCLUSION

Singular Spectrum Analysis (SSA) has appeared over the past 20 years and considered one of the powerful technique for analysing a variety of time series. Although its origins lie in the natural sciences, and the series arisen from such processes, it can be applied in several different fields [6].

Studying master of financial mathematics, financial time series drive my attention continually. The financial world is developing and many of its institutions are studying the analysis of financial time series data. For years different kinds of financial time-series have practically and also theoretically interest for making inferences and predictions. It is desirable to monitor behaviour of stock price and to try to understand the probable development of the prices in the future. Thus, in this paper we have described the methodology of SSA in the context of financial time series and also represented some results of numerical experiments.

Singular Spectrum analysis (SSA) is non-parametric method of time series analysis that decomposes time series into trend and built reconstructed components which upon used for forecasting, It uses linear algebra tools such as eigenvalues and eigenvectors, singular decomposition by creating the trajectory matrix from a time series. In this work we analysed and presented theoretical results on SSA applied to financial time series; applied forecasting method for observation of future behaviour of the initial time series; showed influence of selection of the window length and number of components to forecasting; compared observed value of time series with forecast and illustrated results of numerical experiments by figures. By carrying out experiment with different window lengths we observed that by increasing window length the forecast accuracy gets smaller and the forecast is more exact. On the other hand, we got different forecasting results by changing the number of components. It is observed that, by using sum of even indices reconstructed components we obtain high accuracy with negative value which can't be "good" selection for forecasting algorithm. However, by using first two reconstructed components, odd indices reconstructed components and sum of all reconstructed components we got more exact forecasting.The comparison of forecasting results showed that there is a big influence of selection SSA parameters to forecasting.

It should be mentioned that application of the SSA forecasting algorithm in financial time series has given us some very expected results but has not showed yet its full potential. In the future, work it is important to study forecasting accuracy with even indices reconstructed components. Furthermore, in the future I would like apply of SSA forecasting algorithm different financial time series such as currency changes and compare results with other methods forecasting methods.

REFERENCE

[1] Aldrich, John (2006), "Eigenvalue, eigenfunction, eigenvector, and related terms", in Jeff Miller (Editor), Earliest Known Uses of Some of the Words of Mathematics, retrieved 2006-08-22

[2] Alter O, Brown PO, Botstein D. (2000) Singular value decomposition for genome-wide expression data processing and modeling. *Proc Natl Acad Sci U S A*, **97**, 10101-6.

[3] David Claessen (CERES-ERTI) and Andreas Groth (LMD) CERES-ERTI, Ecole Normale Supérieure, Paris, (Singular Spectrum Analysis) www.environnement.ens.fr

[4] “Diagonalization of Matrices”. (1st ed., pp. 1-2) *Math.okstate.edu*. Retrieved from https://math.okstate.edu/people/binegar/3013-S99/3013-l16.pdf

[5] Golub, G.H., and Van Loan, C.F. (1989) Matrix Computations, 2nd ed. (Baltimore: Johns Hopkins University Press).

[6] H. Hassani and D. Thomakos, A review on singular spectrum analysis for economic and financial time series, Statistics and its Interface, 3 (2010), pp. 377-397

[7] Hassani, H. (2007). Singular Spectrum Analysis: Methodology-and Comparison. *Journal of Data Science* **5** 239–257.

[8] HELM Workbook. (2004) (1st ed., pp. 1-12). Retrieved from http://www.personal.soton.ac.uk/jav/soton/HELM/helm_workbooks.html

[9] Héctor Briceñoa , Claudio M. Roccoa, Enrico Zio, Singular Spectrum Analysis For Forecasting Of Electric Load Demand. 2013. *Aidic.it*. Retrieved from http://www.aidic.it/cet/13/33/154.pdf

[10] Hefferon, Jim (2001), Linear Algebra, Online book, St Michael's College, Colchester, Vermont, USA

[11] "Introduction To Time Series Analysis". *Gauss.stat.su.se*. N.p., 2017. Retrieved from http://gauss.stat.su.se/gu/e/slides/Time%20Series/Introduction%20to%20Time%20Series%20Analysis.pdf

[12] Loan, G., & Loan, V. Singular Value Decomposition (SVD) (1st ed., pp. 1-36). Retrieved from https://www.cs.cmu.edu/~venkatg/teaching/CStheory-infoage/book-chapter-4.pdf

[13] "LU Decomposition -- From Wolfram Mathworld". *Mathworld.wolfram.com*. N.p., 2017. Web. 14 May 2017.

[14] "LU Decomposition - Math-Linux.Com". *Math-linux.com*. N.p., 2017. Web. 14 May 2017.

[15] M. Ghil, M. R. Allen, M. D. Dettinger, K. Ide, D. Kondrashov, M. E. Mann, A.W. Robertson, A. Saunders, Y. Tian, F. Varadi, and P. Yiou, Advanced Spectral Methods For Climatic Time Series, Reviews of Geophysics 40 (2002) 3-1 - 3-41.

[16] N. Golyandina, V. Nekrutkin, and A. Zhigljavsky, Analysis of structure of time series: SSA and related techniques, 2001.

[17] N.Golyandina and A. Zhigljavsky, Singular Spectrum Analysis for Time Series,Springer,New York (2013).

[18] R. Adhikari and R. Agrawal, An introductory study on time series modeling and forecasting, arxiv preprint arxiv:1302.6613, (2013).

[19] Rahim Mahmoudvand, Fatemeh Alehosseini, Paulo Canas Rodrigues, "FORECASTING MORTALITY RATE BY SINGULAR SPECTRUM ANALYSIS". (pp. 193-206) *REVSTAT – Statistical Journal.* 2015 Retrieved from https://www.ine.pt/revstat/pdf/rs150301.pdf

[20] "Singular Value Decomposition - MATLAB Svd - Mathworks United Kingdom". *Se.mathworks.com*.

[21] Staff, Investopedia. "Time Series". *Investopedia*. N.p., 2017. Retrieved from

http://www.investopedia.com/terms/t/timeseries.asp

[22] Todd Young and Martin J. Mohlenkamp, LU Decomposition. (1st ed,. pp. 50-52) http://www.math.ohiou.edu/courses/math3600/

[23] Weisstein, Eric W. "Eigenvalue." From MathWorld--A Wolfram Web Resource. http://mathworld.wolfram.com/Eigenvalue.html

[24] Weisstein, Eric W. "Eigen Decomposition." From *MathWorld*--A Wolfram Web Retrieved from. http://mathworld.wolfram.com/EigenDecomposition.html

[25] Zhigljavsky, A., Hassani, H., and Heravi, H. (2009). Forecasting European Industrial Production with Multivariate Singular Spectrum Analysis, International Institute of Forecasters, 2008–2009 SAS/IIF Grant, https://forecasters.org/pdfs/SAS-Report.pdf

Appendix A

The Matlab script for SSA algorithm used in this study. Results are illustrated in section 4.1-4.3.

```
%SSA algorithm
%Code was used and edited by Aytan Osmanzade for results of the master
thesis
%"Singular spectrum analysis forecasting for financial time series".

%Copyright (c) 2013-2016, Andreas Groth, University of California, Los
Angeles.
%https://se.mathworks.com/matlabcentral/fileexchange/58967-singular-
spectrum-analysis-beginners-guide

% Read Initial data
y=xlsread('tallink2016.xlsx');
X=flipud(y(:,4)); % Initial Data
N=length (X);  %length of Initial Data
L=110;   %window length
t=(1:N); %Initial time period

%Plot Initial data
figure;
plot(t,X);
xlabel('Date')
ylabel('Stock price,EUR')
title('Daily Closing Stock Prices AS Tallink Grupp for 2016 year ')

%Construction of trajectory matrix of the time series
Y=zeros(L,N-L+1); %trajectory matrix
for m=1:N-L+1
  Y(:,m) = X((1:L)+m-1);
end;
figure ;
set(gcf,'name','Trajectory matrix of the time series');
clf;
imagesc(Y); %trajectory matrix
title('Trajectory matrix of time series')
axis square
colorbar

%Calculation of eigenvalues LAMBDA and eigenvectors V of the matrix S
S=Y*Y'/ (N-L+1) ;%calculation of matrix S
[V,LAMBDA] = eig(S);
LAMBDA = diag(LAMBDA); % extract the diagonal elements
[LAMBDA,ind]=sort(LAMBDA,'descend'); % sort eigenvalues
V = V(:,ind);    % V is matrix of eigenvectors (each column is one
eigenvector)

% Plot eigenvalues of the matrix S
figure;
```

```
set(gcf,'name','Eigenvalues of the matrix S')
clf;
plot(LAMBDA,'o-');
title('Eigenvalues of the the matrix S')

%Plot first eight eigenvectors of the matrix S
figure;
set(gcf,'name','Eigenvectors of the matrix ')
clf;
subplot(4,1,1);
plot(V(:,1:2), '-');
legend('1', '2');
subplot(4,1,2);
plot(V(:,3:4), '-');
legend('3', '4');
subplot(4,1,3);
plot(V(:,5:6), '-');
legend('5', '6');
subplot(4,1,4);
plot(V(:,7:8), '-');
legend('7', '8');

% Construction of the Principal Components of the time series
PC = Y'*V; %Principal Components of the time series

%Plot first four principal components
figure;
set(gcf,'name','Principal components ')
clf;

subplot(4,1,1);
plot(PC(:,1),'.-');
ylabel(sprintf('PC %d',1));

subplot(4,1,2);
plot(PC(:,2),'.-');
ylabel(sprintf('PC %d',2));

subplot(4,1,3);
plot(PC(:,3),'.-');
ylabel(sprintf('PC %d',3));

subplot(4,1,4);
plot(PC(:,4),'.-');
ylabel(sprintf('PC %d',4));

%Plot last four principal components
figure;
set(gcf,'name','Principal components ')
clf;

subplot(4,1,1);
plot(PC(:,107),'.-');
ylabel(sprintf('PC %d',107));

subplot(4,1,2);
```

```
plot(PC(:,108),'.-');
ylabel(sprintf('PC %d',108));

subplot(4,1,3);
plot(PC(:,109),'.-');
ylabel(sprintf('PC %d',109));

subplot(4,1,4);
plot(PC(:,110),'.-');
ylabel(sprintf('PC %d',110));

%Construction of Reconstructed Components
R=zeros(N,L);
for m=1:L
  inp=PC(:,m)*V(:,m)'; %invert projection
  inp=inp(end:-1:1,:);
  for n=1:N % anti-diagonal averaging
    R(n,m)=mean( diag(inp,-(N-L+1)+n) );
  end
end;

%Plot first four reconstructed components
figure;
set(gcf,'name','Reconstructed components')
clf;

subplot(4,1,1);
plot(t,R(1:N,1),'r-');
ylabel(sprintf('R %d',1));

subplot(4,1,2);
plot(t,R(1:N,2),'r-');
ylabel(sprintf('R %d',2));

subplot(4,1,3);
plot(t,R(1:N,3),'r-');
ylabel(sprintf('R %d',3));

subplot(4,1,4);
plot(t,R(1:N,4),'r-');
ylabel(sprintf('R %d',4));

%Plot last four reconstructed components
figure;
set(gcf,'name','Reconstructed components')
clf;

subplot(4,1,1);
plot(t,R(1:N,107),'r-');
ylabel(sprintf('R %d',107));

subplot(4,1,2);
plot(t,R(1:N,108),'r-');
ylabel(sprintf('R %d',108));
```

```
subplot(4,1,3);
plot(t,R(1:N,109),'r-');
ylabel(sprintf('R %d',109));

subplot(4,1,4);
plot(t,R(1:N,110),'r-');
ylabel(sprintf('R %d',110));

%% Compare initial time series and reconstructed components
figure;
set(gcf,'name','Initial time serie X and Reconstruction R')
clf;
subplot(3,1,1)
plot(t(1:N),X,'b-');
legend('Initial time series');
subplot(3,1,2)
plot(t,sum(R(:,:),2),'r-');
legend('Complete reconstruction'); %sum of all reconstructed
components
subplot(3,1,3)
plot(t(1:N),X,'b-',t,sum(R(:,1:2),2),'r-');
legend('Initial time series','Reconstruction with Rs 1-2');
%comparison time series with sum of first two reconstructed components
```

Appendix B

The Matlab script for SSA forecasting algorithm used in this study. Results are illustrated in section 4.4.

```
% SSA forecasting algorithm
% Author: Aytan Osmanzade

%Calculation of vector v^2.
kk = find(LAMBDA,1,'last'); % the last non-zero eigenvalues.
v2=sum(V(L,1:kk).^2); % v^2 is calculated by sum squares of the last
components of the eigenvectors.

%Creation of the vector A.
A=zeros(L-1,1);
for k=1:L
    A=A+(V(L,k))*V(1:L-1,k)/(1-v2); %the vector A is calculated by the
formula given in section 3.4
end

%Determine forecasting length
h=Nnew-N;

%Calculaton of the time series F(N+h)
F=sum(R(:, 1:2),2); % the reconstructed time series of the first two
components
```

```
% F=sum(R(:, 1:2:end),2) %calculation for odd indices reconctructed
components
% F=sum(R(:, 2:2:end),2) %calculation for even indices reconctructed
components
for k=1:h %the calculations are made by the formula given in section
3.4
    uu=F(N-L+k+1:N+k-1);
    uuu=sum(A.*uu);
    F=[F; uuu];
end

%Plot the time series F
figure;
plot(F);
title('The time series F ')
```

Appendix C

The Matlab script evaluate the SSA forecast accuracy and compare forecast with observed value of the initial time series Results are illustrated in section 4.4-4.5.

```
% Evaluation of the SSA forecast accuracy and comparison forecast with
observed value of the initial time series
% Author: Aytan Osmanzade

% Read the the iniial time series(with observed value)
ynew=xlsread('tallink2016-2017.xlsx');
Xnew=ynew(:,4); % Initial Data
Nnew=length (Xnew);  %length of Initial Data

% Compare Initial time series (with observed value)and Forecasting
figure;
set(gcf,'name','Initial time series and Forecast')
clf;
subplot(2,1,1)
plot(tnew,Xnew,'b-');
legend('Original time series');
subplot(2,1,2)
plot(tnew,F,'r-');
legend('Complete Reconstruction and Forecast');

% Compare the observed value of the initial time series with Forecast

figure;
set(gcf,'name','The observed value of the initial time series and
Forecast')
clf;
subplot(2,1,1)
plot(tnew(:,261:294),Xnew(261:294,:),'b-');
legend('The observed value of the time series');
subplot(2,1,2)
plot(tnew(:,261:294),F(261:294,:),'r-');
```

```
legend('Forecasting with L=110');

%Calculation of the forecast error and forecast accuracy
e=Xnew(261:294,:)-F(261:294,:);  %forecast error
ISE=sum(e.^2); %forecast accuracy, for L=110; r=2.
```

Appendix D

Close prices (EUR) of “AS Tallink Grupp” stock for 2016 year.

Date	Close price	Date	Close price	Date	Close price	Date	Close price	Date	Close price	Date	Close price	Date	Close price
30/12/2016	0.912	07/11/2016	0.884	12/09/2016	0.873	18/07/2016	0.925	23/05/2016	0.988	28/03/2016	0.929	01/02/2016	0.853
29/12/2016	0.91	04/11/2016	0.881	09/09/2016	0.876	15/07/2016	0.924	20/05/2016	0.993	25/03/2016	0.929	29/01/2016	0.843
28/12/2016	0.907	03/11/2016	0.884	08/09/2016	0.868	14/07/2016	0.92	19/05/2016	0.994	24/03/2016	0.929	28/01/2016	0.85
27/12/2016	0.909	02/11/2016	0.891	07/09/2016	0.878	13/07/2016	0.913	18/05/2016	0.993	23/03/2016	0.925	27/01/2016	0.85
26/12/2016	0.917	01/11/2016	0.905	06/09/2016	0.883	12/07/2016	0.911	17/05/2016	0.985	22/03/2016	0.923	26/01/2016	0.84
23/12/2016	0.917	31/10/2016	0.906	05/09/2016	0.88	11/07/2016	0.909	16/05/2016	0.99	21/03/2016	0.925	25/01/2016	0.85
22/12/2016	0.903	28/10/2016	0.901	02/09/2016	0.877	08/07/2016	0.909	13/05/2016	0.981	18/03/2016	0.927	22/01/2016	0.84
21/12/2016	0.906	27/10/2016	0.907	01/09/2016	0.883	07/07/2016	0.902	12/05/2016	0.986	17/03/2016	0.92	21/01/2016	0.835
20/12/2016	0.896	26/10/2016	0.898	31/08/2016	0.88	06/07/2016	0.905	11/05/2016	0.991	16/03/2016	0.92	20/01/2016	0.84
19/12/2016	0.887	25/10/2016	0.905	30/08/2016	0.88	05/07/2016	0.905	10/05/2016	0.991	15/03/2016	0.918	19/01/2016	0.838
16/12/2016	0.871	24/10/2016	0.9	29/08/2016	0.875	04/07/2016	0.898	09/05/2016	0.985	14/03/2016	0.924	18/01/2016	0.838
15/12/2016	0.858	21/10/2016	0.908	26/08/2016	0.88	01/07/2016	0.904	06/05/2016	0.986	11/03/2016	0.925	15/01/2016	0.835
14/12/2016	0.875	20/10/2016	0.908	25/08/2016	0.885	30/06/2016	0.914	05/05/2016	0.99	10/03/2016	0.92	14/01/2016	0.85
13/12/2016	0.872	19/10/2016	0.91	24/08/2016	0.882	29/06/2016	0.909	04/05/2016	0.99	09/03/2016	0.918	13/01/2016	0.85
12/12/2016	0.871	18/10/2016	0.9	23/08/2016	0.885	28/06/2016	0.962	03/05/2016	0.992	08/03/2016	0.92	12/01/2016	0.85
09/12/2016	0.876	17/10/2016	0.906	22/08/2016	0.877	27/06/2016	0.941	02/05/2016	0.986	07/03/2016	0.927	11/01/2016	0.851
08/12/2016	0.875	14/10/2016	0.908	19/08/2016	0.875	24/06/2016	0.974	29/04/2016	0.99	04/03/2016	0.926	08/01/2016	0.85
07/12/2016	0.875	13/10/2016	0.909	18/08/2016	0.891	23/06/2016	0.974	28/04/2016	0.995	03/03/2016	0.927	07/01/2016	0.857
06/12/2016	0.875	12/10/2016	0.91	17/08/2016	0.895	22/06/2016	0.974	27/04/2016	0.983	02/03/2016	0.92	06/01/2016	0.857
05/12/2016	0.881	11/10/2016	0.912	16/08/2016	0.896	21/06/2016	0.955	26/04/2016	0.98	01/03/2016	0.904	05/01/2016	0.86
02/12/2016	0.877	10/10/2016	0.905	15/08/2016	0.894	20/06/2016	0.946	25/04/2016	0.927	29/02/2016	0.905	04/01/2016	0.843
01/12/2016	0.876	07/10/2016	0.909	12/08/2016	0.87	17/06/2016	0.967	22/04/2016	0.937	26/02/2016	0.905	01/01/2016	0.857
30/11/2016	0.875	06/10/2016	0.902	11/08/2016	0.895	16/06/2016	0.98	21/04/2016	0.935	25/02/2016	0.892		
29/11/2016	0.884	05/10/2016	0.901	10/08/2016	0.949	15/06/2016	1.01	20/04/2016	0.933	24/02/2016	0.888		
28/11/2016	0.87	04/10/2016	0.901	09/08/2016	0.947	14/06/2016	1.02	19/04/2016	0.939	23/02/2016	0.888		
25/11/2016	0.871	03/10/2016	0.906	08/08/2016	0.943	13/06/2016	1.02	18/04/2016	0.939	22/02/2016	0.876		
24/11/2016	0.866	30/09/2016	0.896	05/08/2016	0.957	10/06/2016	1.01	15/04/2016	0.94	19/02/2016	0.878		
23/11/2016	0.869	29/09/2016	0.9	04/08/2016	0.95	09/06/2016	1.03	14/04/2016	0.942	18/02/2016	0.87		
22/11/2016	0.867	28/09/2016	0.892	03/08/2016	0.955	08/06/2016	1.03	13/04/2016	0.94	17/02/2016	0.864		
21/11/2016	0.875	27/09/2016	0.879	02/08/2016	0.952	07/06/2016	1.03	12/04/2016	0.94	16/02/2016	0.865		
18/11/2016	0.875	26/09/2016	0.878	01/08/2016	0.96	06/06/2016	1.02	11/04/2016	0.93	15/02/2016	0.863		
17/11/2016	0.868	23/09/2016	0.878	29/07/2016	0.955	03/06/2016	1.04	08/04/2016	0.93	12/02/2016	0.85		
16/11/2016	0.861	22/09/2016	0.878	28/07/2016	0.93	02/06/2016	1.01	07/04/2016	0.937	11/02/2016	0.85		
15/11/2016	0.866	21/09/2016	0.876	27/07/2016	0.96	01/06/2016	0.999	06/04/2016	0.945	10/02/2016	0.854		
14/11/2016	0.869	20/09/2016	0.877	26/07/2016	0.969	31/05/2016	0.999	05/04/2016	0.937	09/02/2016	0.851		
11/11/2016	0.878	19/09/2016	0.874	25/07/2016	0.967	30/05/2016	0.995	04/04/2016	0.948	08/02/2016	0.862		
10/11/2016	0.88	16/09/2016	0.87	22/07/2016	0.963	27/05/2016	0.995	01/04/2016	0.93	05/02/2016	0.862		
09/11/2016	0.869	15/09/2016	0.88	21/07/2016	0.971	26/05/2016	0.995	31/03/2016	0.939	04/02/2016	0.855		
08/11/2016	0.877	14/09/2016	0.873	20/07/2016	0.962	25/05/2016	1	30/03/2016	0.921	03/02/2016	0.858		
07/11/2016	0.884	13/09/2016	0.885	19/07/2016	0.945	24/05/2016	0.994	29/03/2016	0.924	02/02/2016	0.858		

Appendix E

Forecast results of the close prices of "AS Tallink Grupp" stock with different SSA parameters.

Date	Observed value of the Close price	Forecast L=110, r=2	Forecast error e_(T,h)	Forecast L=110, r=L=110	Forecast error e_(T,h)
03/01/2017	0.919	0.8669	0.0521	0.9060	0.0130
04/01/2017	0.916	0.8671	0.0489	0.9030	0.0130
05/01/2017	0.923	0.8672	0.0558	0.9170	0.0060
06/01/2017	0.922	0.8673	0.0547	0.9170	0.0050
09/01/2017	0.924	0.8674	0.0566	0.9090	0.0150
10/01/2017	0.912	0.8675	0.0445	0.9070	0.0050
11/01/2017	0.92	0.8675	0.0525	0.9100	0.0100
12/01/2017	0.92	0.8676	0.0524	0.9120	0.0080
13/01/2017	0.92	0.8189	0.1011	0.8747	0.0453
16/01/2017	0.915	0.8018	0.1132	0.8579	0.0571
17/01/2017	0.923	0.7832	0.1398	0.8298	0.0932
18/01/2017	0.923	0.7661	0.1569	0.8105	0.1125
19/01/2017	0.926	0.7585	0.1675	0.7915	0.1345
20/01/2017	0.921	0.7485	0.1725	0.7676	0.1534
23/01/2017	0.92	0.7402	0.1798	0.7463	0.1737
24/01/2017	0.92	0.7244	0.1956	0.7319	0.1881
25/01/2017	0.925	0.7173	0.2077	0.7206	0.2044
26/01/2017	0.938	0.7109	0.2271	0.7001	0.2379
27/01/2017	0.939	0.7042	0.2348	0.6865	0.2525
30/01/2017	0.948	0.7087	0.2393	0.6788	0.2692
31/01/2017	0.95	0.7126	0.2374	0.6693	0.2807
01/02/2017	0.957	0.7190	0.2380	0.6748	0.2822
02/02/2017	0.957	0.7282	0.2288	0.6865	0.2705
03/02/2017	0.96	0.7380	0.2220	0.6973	0.2627
06/02/2017	0.96	0.7510	0.2090	0.7097	0.2503
07/02/2017	0.959	0.7632	0.1958	0.7270	0.2320
08/02/2017	0.954	0.7786	0.1754	0.7469	0.2071
09/02/2017	0.963	0.7968	0.1662	0.7710	0.1920
10/02/2017	0.964	0.8136	0.1504	0.7931	0.1709
13/02/2017	0.97	0.8291	0.1409	0.8187	0.1513
14/02/2017	0.975	0.8424	0.1326	0.8373	0.1377
15/02/2017	0.98	0.8490	0.1310	0.8559	0.1241
16/02/2017	0.991	0.8464	0.1446	0.8578	0.1332
17/02/2017	0.976	0.8410	0.1350	0.8622	0.1138
20/02/2017	0.981	0.8350	0.1460	0.8609	0.1201
21/02/2017	0.966	0.8251	0.1409	0.8533	0.1127
22/02/2017	0.977	0.8131	0.1639	0.8370	0.1400
23/02/2017	0.983	0.7994	0.1836	0.8219	0.1611
27/02/2017	0.968	0.7825	0.1855	0.8027	0.1653
28/02/2017	0.957	0.7559	0.2011	0.7710	0.1860
01/03/2017	0.938	0.7375	0.2005	0.7475	0.1905

Date	Observed value of the Close price	Forecast L=40, r=L=110	Forecast error e_(T,h)	Forecast L=10, r=L=110	Forecast error e_(T,h)
03/01/2017	0.919	0.9060	0.0130	0.9060	0.0130
04/01/2017	0.916	0.9030	0.0130	0.9030	0.0130
05/01/2017	0.923	0.9170	0.0060	0.9170	0.0060
06/01/2017	0.922	0.9170	0.0050	0.9170	0.0050
09/01/2017	0.924	0.9090	0.0150	0.9090	0.0150
10/01/2017	0.912	0.9070	0.0050	0.9070	0.0050
11/01/2017	0.92	0.9100	0.0100	0.9100	0.0100
12/01/2017	0.92	0.9120	0.0080	0.9120	0.0080
13/01/2017	0.92	0.1358	0.7842	1.0037	-0.0837
16/01/2017	0.915	0.2759	0.6391	1.0277	-0.1127
17/01/2017	0.923	0.0663	0.8567	1.0707	-0.1477
18/01/2017	0.923	-0.3353	1.2583	1.1385	-0.2155
19/01/2017	0.926	-0.6977	1.6237	1.2283	-0.3023
20/01/2017	0.921	-1.0502	1.9712	1.3153	-0.3943
23/01/2017	0.92	-1.5281	2.4481	1.4204	-0.5004
24/01/2017	0.92	-2.0773	2.9973	1.5468	-0.6268
25/01/2017	0.925	-2.5365	3.4615	1.6775	-0.7525
26/01/2017	0.938	-3.0983	4.0363	1.8327	-0.8947
27/01/2017	0.939	-3.8802	4.8192	2.0110	-1.0720
30/01/2017	0.948	-4.3120	5.2600	2.2070	-1.2590
31/01/2017	0.95	-5.2036	6.1536	2.4229	-1.4729
01/02/2017	0.957	-5.8108	6.7678	2.6688	-1.7118
02/02/2017	0.957	-6.4120	7.3690	2.9432	-1.9862
03/02/2017	0.96	-6.9164	7.8764	3.2463	-2.2863
06/02/2017	0.96	-7.4248	8.3848	3.5853	-2.6253
07/02/2017	0.959	-7.9249	8.8839	3.9627	-3.0037
08/02/2017	0.954	-8.1219	9.0759	4.3809	-3.4269
09/02/2017	0.963	-8.3550	9.3180	4.8458	-3.8828
10/02/2017	0.964	-8.1947	9.1587	5.3632	-4.3992
13/02/2017	0.97	-7.7807	8.7507	5.9368	-4.9668
14/02/2017	0.975	-7.5134	8.4884	6.5729	-5.5979
15/02/2017	0.98	-6.4902	7.4702	7.2793	-6.2993
16/02/2017	0.991	-5.5779	6.5689	8.0630	-7.0720
17/02/2017	0.976	-4.0351	5.0111	8.9319	-7.9559
20/02/2017	0.981	-2.1444	3.1254	9.8957	-8.9147
21/02/2017	0.966	0.0270	0.9390	10.9647	-9.9987
22/02/2017	0.977	2.6954	-1.7184	12.1497	-11.1727
23/02/2017	0.983	5.5200	-4.5370	13.4637	-12.4807
27/02/2017	0.968	9.0553	-8.0873	14.9207	-13.9527
28/02/2017	0.957	12.5890	-11.6320	16.5360	-15.5790
01/03/2017	0.938	16.8475	-15.9095	18.3266	-17.3886

Non-exclusive licence to reproduce thesis and make thesis public

I, Osmanzade Aytan Ali (date of birth: 17.12.1990),

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to:
   1.1. reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and
   1.2. make available to the public via the web environment of the University of Tartu, including via the DSpace digital archives until expiry of the term of validity of the copyright,

"Singular spectrum analysis forecasting for financial time series", supervised by Peep Miidla

2. I am aware of the fact that the author retains these rights.
3. I certify that granting the non-exclusive licence does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Tartu, **16.05.2017**