

DOI: 10.17323/2587-814X.2023.3.87.100

Forecasting financial time series using singular spectrum analysis*

Anna V. Zinenko

E-mail: anna-z@mail.ru

Siberian Federal University

Address: 79, Svobodny Prospect, Krasnoyarsk 660041, Russia

Abstract

Financial time series are big arrays of information on quotes and trading volumes of shares, currencies and other exchange and over-the-counter instruments. The analysis and forecasting of such series has always been of particular interest for both research analysts and practicing investors. However, financial time series have their own features, which do not allow one to choose the only correct and well-functioning forecasting method. Currently, machine-learning algorithms allow one to analyze large amounts of data and test the resulting models. Modern technologies enable testing and applying complex forecasting methods that require volumetric calculations. They make it possible to develop the mathematical basis of forecasting, to combine different approaches into a single method. An example of such a modern approach is the Singular Spectrum Analysis (SSA), which combines the decomposition of a time series into a sum of time series, principal component analysis and recurrent forecasting. The purpose of this work is to analyze the possibility of applying SSA to financial time series. The SSA method was considered in comparison with other common methods for forecasting financial time series: ARIMA, Fourier transform and recurrent neural network. To implement the methods, a software algorithm in the Python language was developed. The method was also tested on the time series of quotes of Russian and American stocks, currencies and cryptocurrencies.

*The article is published with the support of the HSE University Partnership Programme

Keywords: non-stationary time series, forecasting, singular spectrum analysis, error metrics

Citation: Zinenko A.V. (2023) Forecasting financial time series using singular spectrum analysis. *Business Informatics*, vol. 17, no. 3, pp. 87–100. DOI: 10.17323/2587-814X.2023.3.87.100

Introduction

Datasets on the market prices of various financial instruments, such as stocks, currencies, derivatives and precious metals can be referred to as financial time series. Time series of this type have certain peculiarities which have to be taken into account in the application of various forecasting methods. The background investment theories, such as the Markowitz and SHARP model or the Black-Scholes option pricing model [1], implied that market time series were random and followed the law of normal distribution. However, simultaneous investigations carried out by Mandelbrot [1, 2] and by Peters [3, 4] proved price time series to be non-random and, consequently, non-stationary; thus, for their forecast, the application of the methods based on the assumption of random processes would not give any adequate result.

The following peculiarities of financial time series can be specified.

1. As is indicated above, time series of market prices are non-stationary. This means that the average value and variance are unstable within the interval under study. Therefore, before applying to these time series the methods which function well with random processes, they have to be transformed to the stationary form. This can be done, for example, by differencing, as is suggested by the ARIMA model.
2. Financial time series are persistent. This implies that subsequent parameters strongly depend on the previous ones. Close to this situation is the sensitivity to the initial conditions, which is typical for chaos. The Hurst exponent [3] allows one to determine whether the time series is persistent. If this exponent fluctuates within the limits from 0.5 to 1, then this time series is persistent. Hurst referred

to the processes of this type as “long memory processes.” The calculation of the Hurst exponent for numerous time series confirms their persistent character [4, 5].

3. There is a large amount of open access information on market time series, which makes it possible to apply methods of machine learning and data analysis to these time series (including deep learning which requires a big volume of training and test sets). Analysts and traders successfully use trading algorithms based on artificial neural networks and other machine learning methods. Financial data are available for various time intervals, from minutes to weeks. On the one hand, dealing with minute data for several years one can obtain a huge amount for training a neural network. However, on the other hand, such sets can be rather noisy [6].
4. Financial time series are non-differentiable, though continuous. If we consider a graph of stock market quotes, it is possible to see that it is not smooth. This means that it is impossible to draw a tangent to it, and thus, to calculate a derivative in any point. An example of a non-differentiable continuous function is the Weierstrass function [1], which resembles a no-trend stock exchange series. This peculiarity has to be taken into account in the application of certain methods, for example, the Fourier transform, which decomposes a time series into a sum of trigonometric functions. In the case of applying the Fourier transform to non-differentiable series, the original series has to be smoothed, for example, by a moving average.

All the above mentioned features should be considered when choosing methods to forecast financial time series. For example, long memory can be taken into account by certain types of recurrent neural networks, such as LSTM neural networks. In the process

of learning, they are capable of choosing from the past data those which have the most significant influence on the values being forecast.

To test the forecast adequacy and to compare the forecasting methods, the time series is to be divided into the training and test components. By using the training component, the algorithm learns (for example, ARIMA parameters, Fourier coefficients, regression coefficients are chosen etc.). Then, the model obtained for the training time series is applied for forecasting and the predictive time series is compared with the test series using metrics [7].

All the internal metrics are based on estimating the variance between actual and predictive values. The mean absolute error (MAE) shows the average variance between actual and predictive values; the mean squared error (MSE) is the average of the squared differences; and the mean absolute percentage error (MAPE) is the average of the relative differences. MAPE is calculated by the following formula:

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \cdot 100 \right|, \quad (1)$$

where n is the number of observations in the test set;
 y_i is the actual value of the parameter in the test set;
 \hat{y}_i is the predictive value of the parameter.

In testing the suggested method, use was made of the MAPE metric since it allows us to estimate the deviation of the forecast data from the actual ones in relative terms, i.e. it shows the deviation of the forecast from the fact in percentage.

1. Materials and methods

In this study, a forecast of a financial time series was made using singular spectrum analysis (SSA), i.e. the ‘‘Caterpillar’’ method. As is the case with the Fourier transform, SSA decomposes the original time series into a sum of components. However, in the Fourier analysis, the time series (the summands) are periodic functions of different frequency and amplitude, while SSA decomposes the original series into trending, periodic and noise elements

[8, 9]. Thus, SSA takes into account such peculiarities of the financial time series as non-stationarity and non-differentiability.

The caterpillar method is a variety of the SSA analysis independently developed in the USSR at the end of the 80s. At present, the study by Golyandina ‘‘Caterpillar Method – SSA Analysis of Time Series’’ [10] gives the most comprehensive description of the method. It is worth noting that foreign authors also refer to the study by Golyandina as the original source [11, 12]. Other studies devoted to SSA analysis also belong to the Russian authors Leontyeva [13] and Danilov [14]. The authors of research works and guidebooks specify two problems in implementing the method: the choice of the caterpillar length and the choice of the main components. The second problem can be solved by considering the contribution of each component to the total variance; however, recommendations concerning the caterpillar length are more likely to have a heuristic character [15].

The essence of the method of singular spectrum analysis is that the original time series is transformed into a matrix, and then, the matrix is divided into components by singular value decomposition (the main components method is used here). The next step depends on the aims of the analysis: either the components are decomposed into the trending, periodic and noise elements and they are used in the analysis and in forecasting (with the noise elements being removed), or the main components are chosen and they are used to continue the series to the length of the step which was set upon the formation of the initial matrix. In this study, the second approach is used. The SSA algorithm is the following.

The time series being analyzed has the length n . The caterpillar length L is chosen, $2 \leq L \leq n/2$, and the matrix X is constructed, as obtained by the shift of each following column by one value of the dimension $L \times (n - L + 1)$:

$$\begin{matrix} x_1 & x_2 & \cdots & \cdots & x_{n-L+1} \\ x_2 & x_3 & \cdots & \cdots & x_{n-L+2} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_L & x_{L+1} & \cdots & \cdots & x_n \end{matrix} \quad (2)$$

Then, the singular value decomposition of the matrix X was performed:

$$X = U \sigma V^T, \tag{3}$$

where U is the left singular vector;
 σ is the diagonal matrix of the values;
 V^T is the right singular vector.

The obtained elements of the vectors are sorted in descending order of the eigenvalues. The initial matrix X is decomposed into L elementary matrices:

$$X_j = \sum_{j=1}^d \sigma_j U_j V_j^T, \quad j = \overline{1, L}, \tag{4}$$

where σ_j is the j -th eigenvalue;
 $U_j V_j^T$ are the left and right singular vectors corresponding to the eigenvalue;
 d is the rank of the matrix X .

The matrices X_j have the dimension $L \times (n - L + 1)$. In order to transform the given matrices to the one-dimensional form, diagonal averaging is used. Each row of the matrix is shifted by the value $i = 1 \dots n - L + 1$, and then, the mean values are calculated in the column which represents the calculated values of the j -th component of the original time series. The procedure of diagonal averaging is shown by formula (5):

$$\begin{array}{cccccccc}
 x_{11} & x_{12} & \dots & \dots & x_{1(n-L+1)} & & & \\
 & x_{21} & x_{22} & \dots & \dots & x_{2(n-L+1)} & & \\
 & & \dots & \dots & \dots & & \dots & \dots \\
 & & & & & & x_{L1} & x_{L2} & \dots & \dots & x_{L(n-L+1)} \\
 \bar{x}_1 & \bar{x}_2 & \dots & \dots & \dots & & \dots & \dots & \dots & \dots & \bar{x}_n
 \end{array} \tag{5}$$

Thus, the original time series is decomposed to the sum of L series. Further, the obtained series are analyzed for trending, periodicity or noise, or the main components are selected, namely the components of the original time series which most significantly influence its dynamics. It is possible to analyze the character of the components, for example, with the

help of heat maps; however, the goal of the present study is forecasting rather than analysis, and thus, we used only the method of the main components, as is indicated above. The contribution of each obtained component to the total variance is calculated by the formula:

$$\frac{\sigma_j^2}{\sum_{j=1}^d \sigma_j^2}, \tag{6}$$

where σ_j^2 is the square of the j -th eigenvalue.

Figure 1 shows the variances of 10 components. As is clear from the figure, the first eigenvalue makes the most significant contribution to the total variance. Let us recall that the eigenvalues and their corresponding left and right singular vectors are sorted in descending order of the eigenvalues. Therefore, as is indicated in Fig. 1, only the series X_1 can be chosen for the forecast.

The final step of the analysis is the actual forecast using the selected main components. Note that if there are more than one component which have a significant impact on the variance, they are to be summed up. In the present study, use was made of the method of recurrent forecasting. For this purpose, the last $2L + 1$ elements of the obtained time series were used and the following system of linear equations was constructed:

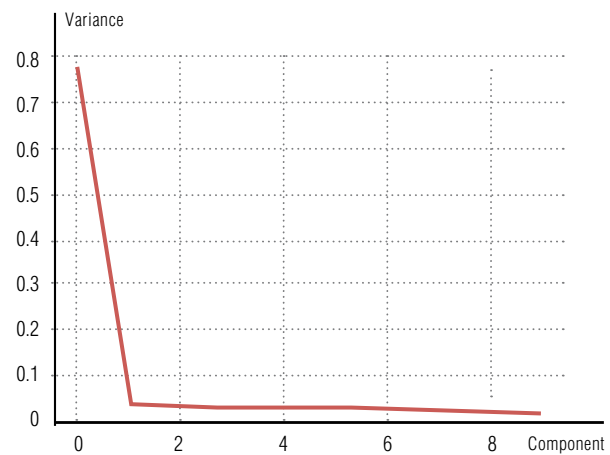


Fig. 1. Contribution of the component variances.

$$\begin{aligned}
 a_1 X_{n-2L+1} + a_2 X_{n-2L+2} + \dots + a_L X_{n-L-1} &= X_{n-L} \\
 a_1 X_{n-2L+2} + a_2 X_{n-2L+3} + \dots + a_L X_{n-L} &= X_{n-L+1} \\
 \dots & \\
 a_1 X_{n-L+1} + a_2 X_{n-L+2} + \dots + a_L X_{n-1} &= X_n.
 \end{aligned}
 \tag{7}$$

Solving this system in terms of the coefficients a , one obtains a matrix of coefficients which is then substituted into the matrix constructed in a manner analogous to formula (7) from the final L time series values and thus, obtains predictive L values:

$$\begin{aligned}
 a_1 X_{n-L+1} + a_2 X_{n-L+2} + \dots + a_L X_n &= X_{n+1} \\
 a_1 X_{n-L+2} + a_2 X_{n-L+3} + \dots + a_L X_{n+1} &= X_{n+2} \\
 \dots & \\
 a_1 X_{L+1} + a_2 X_{L+2} + \dots + a_L X_{n+L-1} &= X_{n+L}.
 \end{aligned}
 \tag{8}$$

Also, for comparison of the results obtained by SSA, forecasting by other methods was performed in this study, namely using ARIMA (autoregressive moving average model), the Fourier transform and the recurrent neural network. The ARIMA model was chosen, being the most popular for forecasting financial time series. The Fourier transform was chosen since it applies a principle somewhat similar to SSA: the original time series is also decomposed into a sum of several time series (only in the case of Fourier analysis all these series are periodic). As concerns the recurrent neural network, this method was chosen due its perspectives and due to the fast development of machine learning methods, in particular, deep learning.

The ARIMA model has three parameters: the autoregression order p , the order of differencing d and the moving average order q . The order of differencing is determined by the Dickey-Fuller test. The autoregression order is determined by the autocorrelation plot of the series levels, in which the time lags are shown on the X -axis, while the values of the correlation coefficient between the levels corresponding to the lag are given on the Y -axis. The autoregression order is chosen to be equal to such a time lag at which the correlation coefficient has the last maximum

value which is different from zero. The moving average order is chosen in the same way; though, instead of the autocorrelation coefficients, partial autocorrelation coefficients are calculated. Partial autocorrelation differs from autocorrelation by not taking into account the impact of the levels which are located between the current level and the level which has a single time lag. It is obvious that in the case of a single lag, autocorrelation and partial autocorrelation coincide.

To do ARIMA forecasting in Python, it is sufficient to determine the parameters p, d, q of the model. Formally, the ARIMA model is described as follows:

$$\Delta^d y_t = \sum_{i=1}^p \alpha_i \Delta^d y_{t-i} - \sum_{j=1}^q \beta_j \varepsilon_{t-j}, \tag{9}$$

where Δ^d is the difference of d -order required to achieve stationarity;

- α_i are the p -order autoregression coefficients;
- β_j are the q -order moving average coefficients;
- ε_{t-j} are the moving average forecasting errors.

The coefficients α_i and β are estimated and substituted into the forecast.

The Fourier transform decomposes the periodic function into a sum of sinusoids and cosine curves with known frequencies, amplitudes and phases. In general terms, the Fourier transform is represented by formula (10):

$$y_t = \bar{y}_t + \sum_{i=1}^{N/2} (\alpha_i \cos \omega_i t + b_i \sin \omega_i t), \tag{10}$$

- where y_t is the transformed value of the time series;
- \bar{y}_t is the average value of the original time series;
- ω_i is the frequency of the i -th harmonic (the first frequency corresponds to the period of the function, with the other ones being multiples of it);
- α_i, β_i are the coefficients to be estimated.

The coefficients of the Fourier series are calculated using the following formulas:

$$a_1 = \frac{2}{N} \sum_{t=0}^{2\pi(N-1)/N} y_t \cos t, \quad b_1 = \frac{2}{N} \sum_{t=0}^{2\pi(N-1)/N} y_t \sin t \tag{11}$$

for the first harmonic,

$$a_2 = \frac{2}{N} \sum_{t=0}^{2\pi(N-1)/N} y_t \cos 2t, b_2 = \frac{2}{N} \sum_{t=0}^{2\pi(N-1)/N} y_t \sin 2t \quad (12)$$

for the second harmonic and so on. Usually, the first two or three harmonics are sufficient for forecasting. The Fourier coefficients obtained by formulas (11) and (12) are substituted into the equation (10), and forecasting is performed using this equation. In this work, for forecasting by the Fourier transform method, use was made of the code in the Python programming language developed by the author.

The time series forecasting algorithm using a **recurrent neural network** can be represented as follows.

1. The original time series of the length n is transformed into a matrix in which the rows correspond to the step L for building a forecast. An array of dimension $(L \times (n - L + 1))$ is fed to the input of the recurrent layer. Also, m is set, which is a step describing how many periods ahead the forecast is made.
2. Inside the recurrent layer, the given array is processed by the activation function, and the output is an array of the dimension specified by the user. If this layer is the last one, then the dimension of the output array is $(m \times 1)$.
3. The final solution is compared with the actual data. The loss function is set (the difference between the actual data and those predicted by the neural network), and using the optimization function, the error backpropagation algorithm is implemented. The error backpropagation algorithm changes the weights given randomly at the stage of the forward operation of the neural network in such a way as to minimize the loss function. Large data arrays for “fitting” the weights are divided into packages (batches), i.e., the optimizer changes the weights after the package is sent rather than after each signal sent. The package size is set upon constructing the neural network and it is usually a power of two. The number of epochs determines the number of “runs” of the neural network for successful learning. The training quality is determined by the loss function on the training data and by the error metric on the test data.

2. Results and discussion

For the practical application of SSA, an algorithm was developed in Python using the numpy, pandas, matplotlib, sclearn libraries. Data on foreign financial time series were taken from the Yahoo! Finance website using the Python yfinance library, and data on domestic stock quotes were obtained using the Python apimox library.

For the analysis, daily stock quotes of the top Russian and American companies were chosen for the time period from June 2022 to March 2023. A total of 30 companies were analyzed: 15 of them Russian and 15 American. Data for 110 days were divided into training and test sets. The training set included 100 values, and the test set had 10 values. Since the forecast was made for the period L – the caterpillar length; then, accordingly, the parameter L was chosen equal to 10. Thus, the original matrix X had the dimension 10×91 , and as a result of the singular value decomposition, it was decomposed into 10 matrices of the same dimension. These matrices were transformed into one-dimensional arrays of dimension 100, and those were selected from them whose matrix eigenvalues make the greatest contribution to the variance. Then, the obtained predictive values were compared with the test set using the MAPE metric. *Table 1* presents the MAPE values of the SSA-forecast for the analyzed stocks.

The average forecast error was 5.54%, including 4.62% for domestic stocks and 6.46% for US stocks. Note that there is a strong outlier: for the company JPMorgan Chase & Co., the error turned out to be significant (26%). For the Russian companies, there is a rather large error only for RUSAL, 11% (which is quite an acceptable result for the forecast). Regarding the stocks of US companies, there is also a significant error for Mastercard (17%). The errors are also quite large for United Health (9.5%) and Advanced Micro Devices (11%). The best forecasts were obtained for Norilsk Nickel (the error being 0.5%), MTS (with the error of 1.7%) and McDonald’s (1.2%). *Fig. 2* shows several plots illustrating the SSA method, with both “bad” and “good” forecasts visualized.

Table 1.

MAPE of the SSA method for stocks

Russia	MAPE	USA	MAPE
Polymetal	7.00%	Amazon	5.00%
Polyus Gold	5.30%	Apple	4.70%
Mechel	7.80%	American Express Company	6.20%
MMC Norilsk Nickel	0.50%	Tesla	2.60%
Yandex	5.20%	Advanced Micro Devices	11.00%
Aeroflot - Russian Airlines	2.90%	Pfizer	2.10%
VTB Bank	4.60%	Netflix	2.12%
Magnit	2.90%	Microsoft	2.00%
Alrosa	5.20%	Mastercard	17.00%
Tinkoff Group	6.10%	Visa	2.10%
RUSAL	11.00%	Starbucks	2.30%
Novatek	2.40%	JPMorgan Chase & Co.	26.00%
Surgutneftegas	3.20%	McDonald's	1.20%
MTS	1.70%	Boeing	3.10%
Severstal	3.50%	United Health	9.50%

To compare the SSA analysis with ARIMA, the Fourier transform and recurrent neural network, a forecast was made for the same stocks over the same time period. The p and d parameters of the ARIMA model for all the stocks were 2 and 1, respectively, while the q parameter varied, depending on the auto-correlation plot. The Fourier transform was performed in three harmonics. The recurrent neural network was constructed using the Keras and Tensor Flow Python libraries. Since the analyzed interval was very small for the neural network, one RNN layer was chosen to avoid overtraining. The activation function, loss function and optimizer function were chosen based on the parameters recommended in [6] for predictive recurrent neural networks. The minimum package size was chosen, i.e. 2, and the number of epochs was chosen such that the loss function and the error (mean abso-

lute error and mean absolute error in percent, respectively) stopped changing. Most experiments used 20 epochs. The forecast was made for 10 days ahead similar to the SSA method.

As is seen from the *Table 2*, the ARIMA method significantly outperformed other methods, including SSA, in terms of the forecast accuracy. However, the SSA analysis showed more accurate results than the Fourier transform and recurrent neural network, and regarding the shares of Norilsk Nickel, MTS and McDonald's it is rather comparable to ARIMA. In general, the Fourier transform also showed good results, with error outliers being present only for the shares of Mechel, Advanced Micro Devices and Pfizer. As concerns the recurrent neural network, it does not work for the stocks for a short period, which was confirmed by the unacceptable size of errors.

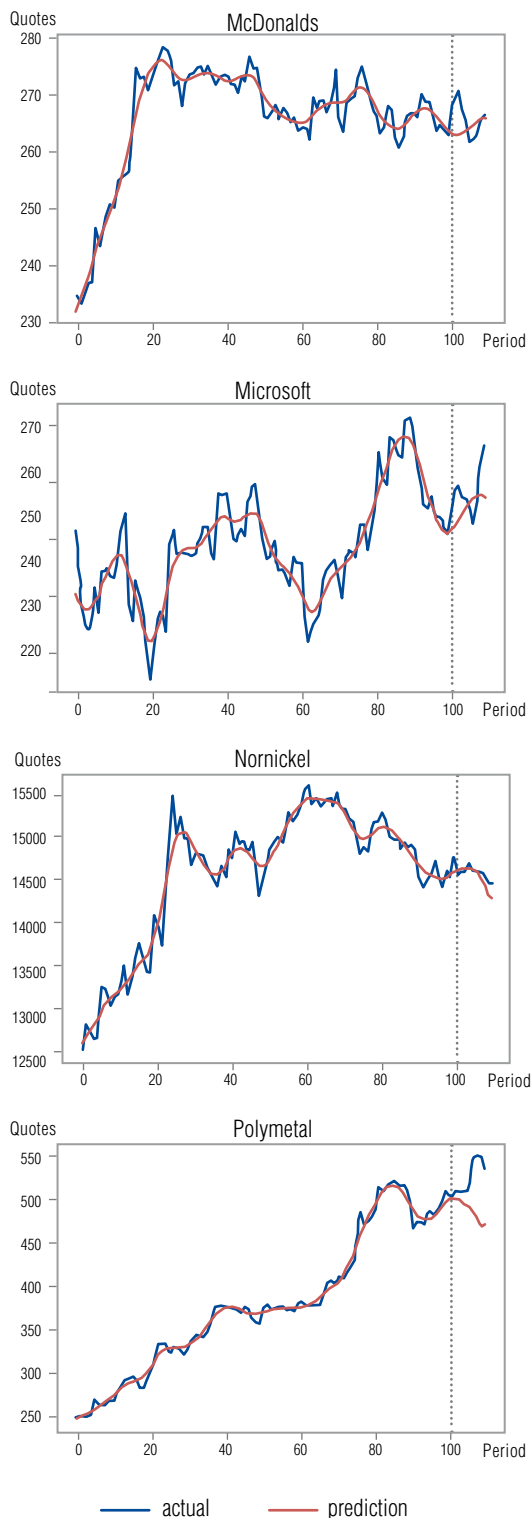


Fig. 2. Visualization of the SSA forecast.

Next, singular spectrum analysis as compared with other methods was applied to the foreign exchange market and cryptocurrency market, which are more dynamic than the stock market. The time period was the same – from June 2022 to March 2023. Ten currency pairs and ten cryptocurrency quotes against the US dollar were taken. The average absolute error in percent for the compared methods is given in *Table 3*.

The SSA method did not show very good results both for fiat and crypto currencies. While for the cryptocurrencies, there were outliers only for BNB and Tron currencies; for the fiat currency pairs, the method showed a large error in four cases out of ten. The ARIMA method worked just as well on the currencies as it did on the stocks, and the Fourier transform and recurrent neural network significantly improved the forecast accuracy. But, the recurrent neural network forecast still remained unsatisfactory. Note that all the considered methods had worse results on the cryptocurrencies than on the fiat ones. This can be due to the fact that there are still loopholes for arbitrageurs since the crypto market is rather young. *Figure 3* shows the forecast plots for some fiat and cryptocurrencies, as obtained by the considered methods.

To solve the problem of the poor SSA forecast for both fiat and cryptocurrencies, we doubled the time interval. New training and test sets for the same currencies were 200 and 20 (the length of the caterpillar, accordingly, was also chosen to be 20). The average absolute error in percent under the new conditions is shown in *Table 4*.

With the increase in the time interval, the results of the SSA forecast improved significantly, while the accuracy of the ARIMA forecast remained almost unchanged, and that of the Fourier forecast was even worse. The recurrent neural network significantly improved the forecast accuracy, though significant outliers remained for two fiat currencies and three cryptocurrencies. The best accuracy was still obtained with the ARIMA model. *Figure 4* shows the forecasts for the currencies over a long time period.

Table 2.

**MAPE of the methods ARIMA,
Fourier transform, RNN**

Russia	MAPE, ARIMA	MAPE, Fourier	MAPE, RNN
Polymetal	1.70%	27.60%	95.77%
Polyus Gold	1.50%	17.00%	99.79%
Mechel	1.80%	22.20%	19.50%
MMC Norilsk Nickel	0.50%	1.50%	54.83%
Yandex	1.20%	3.20%	95.51%
Aeroflot	0.70%	8.80%	5.86%
VTB	0.80%	1.50%	3.60%
Magnit	0.40%	3.30%	99.05%
Alrosa	0.70%	2.30%	29.19%
Tinkoff Group	1.00%	3.90%	98.34%
RUSAL	0.70%	5.00%	2.65%
Novatek	0.80%	2.60%	95.85%
Surgutneftegas	0.80%	3.50%	2.19%
MTS	1.00%	8.40%	81.83%
Severstal	1.10%	18.40%	95.12%
Average for Russian stocks	0.98%	8.61%	58.61%
Amazon	1.50%	3.10%	56.39%
Apple	1.60%	6.40%	69.55%
American Express Company	2.30%	8.50%	70.46%
Tesla	3.10%	5.90%	76.90%
Advanced Micro Devices	2.00%	17.00%	34.24%
Pfizer	1.00%	17.10%	8.81%
Netflix	1.70%	3.00%	83.86%
Microsoft	1.30%	5.90%	81.80%
Mastercard	1.40%	3.80%	86.55%
Visa	1.30%	4.80%	78.09%
Starbucks	1.10%	3.50%	53.21%
JPMorgan Chase & Co.	2.70%	4.10%	64.73%
McDonald's	0.80%	2.00%	83.24%
Boeing	2.10%	10.80%	72.31%
United Health	0.70%	9.40%	91.70%
Average for US stocks	1.64%	7.18%	67.46%
Total average	1.31%	7.90%	63.03%

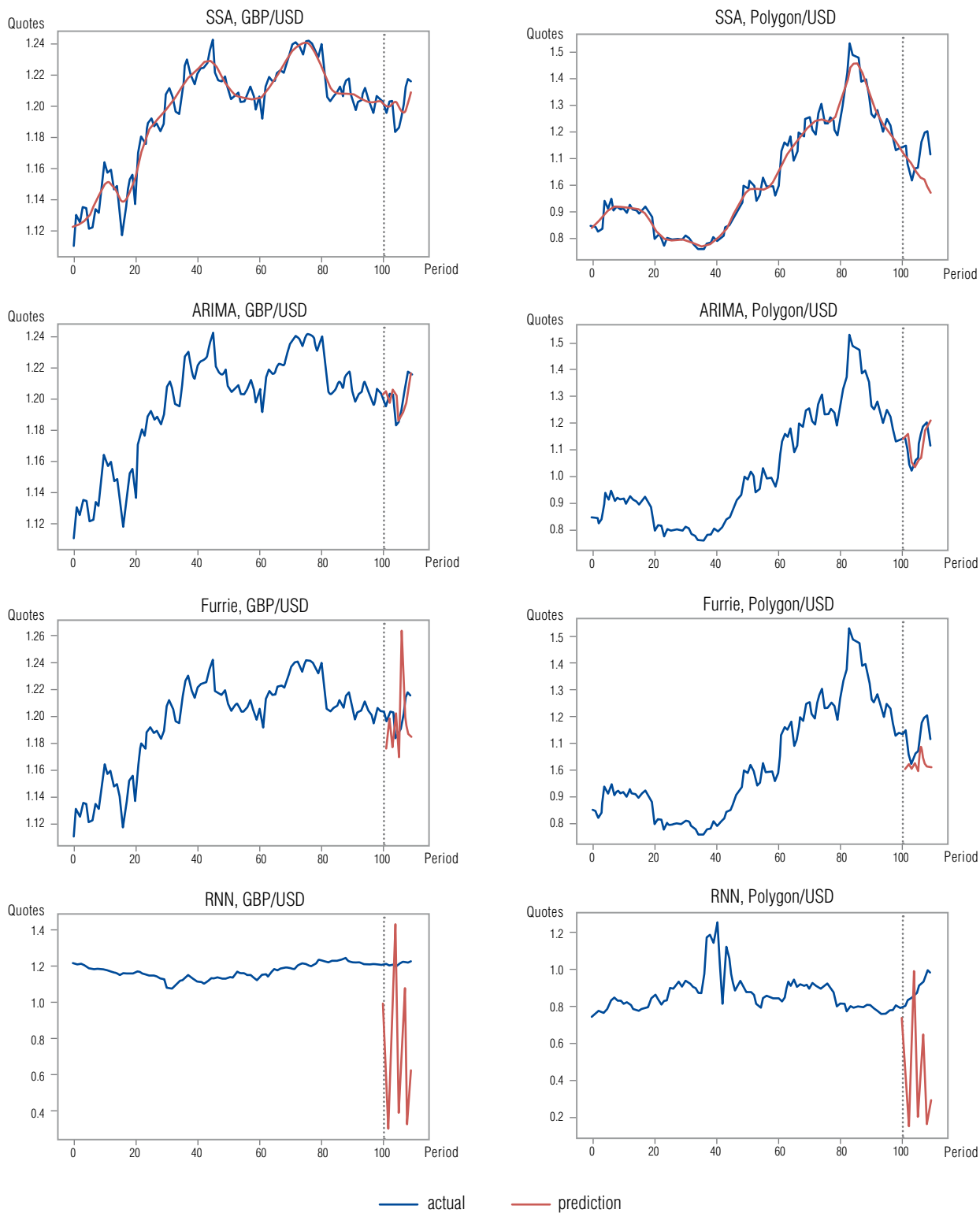


Fig. 3. Visualization of the currency forecasts by SSA, ARIMA, Fourier transform and RNN.

Table 3.

**MAPE of SSA, ARIMA, Fourier transform
and RNN for currencies**

Currency	MAPE, SSA	MAPE, ARIMA	MAPE, Fourier	MAPE, RNN
Euro / US Dollar	59.00%	0.50%	2.70%	1.89%
Pound / US Dollar	0.70%	0.60%	2.10%	2.22%
US Dollar / Yuan	3.70%	0.50%	1.70%	1.47%
US Dollar / Rouble	37.90%	0.30%	12.20%	30.46%
US Dollar / Yen	40.20%	0.60%	1.80%	69.06%
US Dollar / Hong Kong Dollar	43.60%	0.00%	1.80%	0.33
US Dollar / South African Rand	8.30%	0.80%	5.00%	2.27%
Australian Dollar / US Dollar	0.90%	0.60%	1.90%	0.33%
US Dollar / Mexican Peso	1.70%	1.10%	5.20%	1.36%
New Zealand Dollar / US Dollar	2.00%	0.70%	1.90%	2.42%
Average for fiat currencies	16.38%	0.57%	3.63%	15.90%
Bitcoin	6.80%	3.10%	9.80%	99.90%
Ethereum	8.20%	3.10%	8.30%	98.75%
Binance Coin	47.3%	1.80%	4.00%	6.57%
Polygon	8.10%	4.00%	8.00%	6.25%
Lightcoin	12.30%	5.70%	8.40%	74.24%
Ripple	1.90%	1.90%	2.50%	6.29%
Polkadot	6.40%	3.50%	4.10%	5.70%
Chainlink	7.40%	3.30%	4.60%	9.44%
Avalanche	17.40%	4.70%	4.8%	11.13%
Tron	45.50%	4.00%	6.70%	4.52%
Average for cryptocurrencies	16.13%	3.51%	6.08%	34.82%
Total average	16.26%	2.04%	4.85%	25.36%

Conclusion

In this paper, a method of forecasting time series, namely the SSA analysis (caterpillar method) was considered. The method was implemented by developing an algorithm in the Python language, and then, was tested on 30 time series of Russian and US stock quotes, as well as on 20 fiat and cryptocurrencies against the US dollar. For comparison, three forecast-

ing methods were taken: ARIMA, the Fourier transform and recurrent neural network. In all the cases, the SSA analysis, except for the currencies with a short time period, showed the second most accurate result after the ARIMA method. For some securities and shares, the SSA error is comparable to that of ARIMA. At the same time, an increase in the time interval significantly improved the SAA results, while the ARIMA results remained unchanged.

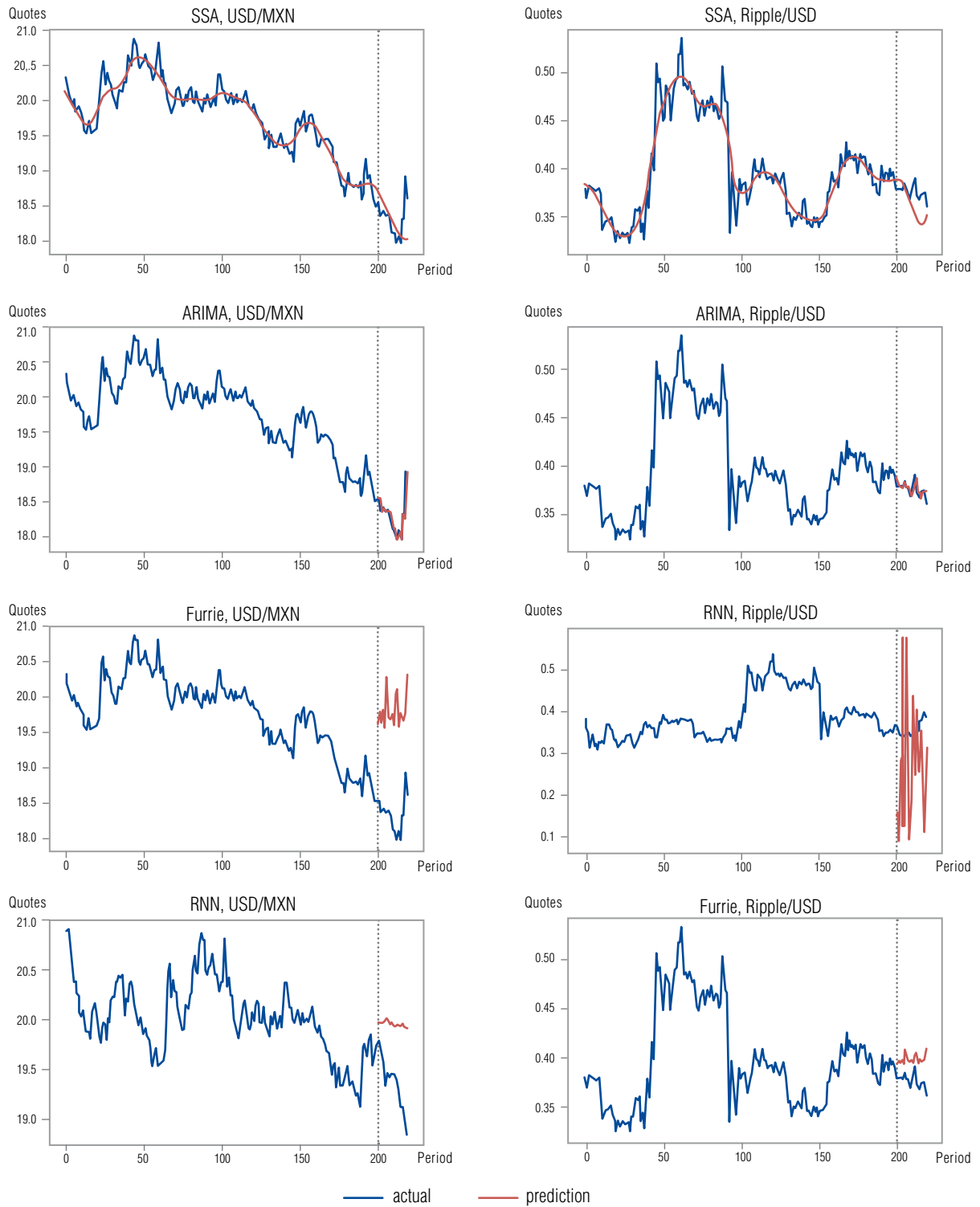


Fig. 4. Visualization of the forecasts for the currencies obtained by SSA, ARIMA, Fourier transform and RNN (long time period).

Table 4.

MAPE of the SSA method and recurrent neural network for currencies (long period)

Currency	MAPE, SSA	MAPE, ARIMA	MAPE, Fourier	MAPE, RNN
Euro / US dollar	2.60%	0.40%	3.00%	2.44%
Pound / US dollar	8.20%	0.50%	1.20%	2.62%
US dollar / Yuan	0.90%	0.40%	1.10%	2.40%
US dollar / Rouble	1.50%	1.10%	16.30%	50.53%
US dollar / Yen	6.50%	0.50%	1.50%	78.31%
US dollar / Hong Kong dollar	1.00%	0.00%	0.90%	0.22%
US Dollar / South African Rand	1.80%	0.60%	6.60%	2.94%
Australian Dollar / US Dollar	3.10%	0.50%	1.70%	0.38%
US Dollar / Mexican Peso	1.10%	0.70%	8.1%	1.53%
New Zealand Dollar / US Dollar	3.20%	0.50%	1.10%	2.99%
Average for fiat currencies	3.16%	0.52%	4.48%	14.44%
Bitcoin	9.90%	2.30%	13.30%	99.83%
Ethereum	48.80%	2.40%	9.90%	97.76%
Binance Coin	3.20%	1.50%	3.80%	7.22%
Polygon	16.50%	3.70%	20.50%	7.23%
Lightcoin	12.40%	3.90%	19.40%	50.07%
Ripple	4.40%	1.50%	6.10%	8.88%
Polkadot	19.70%	3.20%	5.80%	6.32%
Chainlink	10.80%	2.90%	5.20%	8.55%
Avalanche	7.40%	4.00%	6.2%	18.19%
Tron	7.00%	2.70%	9.80%	4.73%
Average for cryptocurrencies	14.01%	2.81%	10.02%	30.88%
Total average	8.58%	1.67%	7.25%	22.66%

It can be concluded that although the SSA analysis shows a lower forecast accuracy than ARIMA generally accepted in the analysis of financial time series, it can be applied both to stocks and to other financial instruments, even to cryptocurrencies which are volatile. It can be used to confirm the ARIMA results, as well as separately as a forecasting method. Note that for the analysis of a large set of stocks, the SSA method is more

convenient than ARIMA, since ARIMA requires recalculation at least of the order of the moving average for each time series, while a single main component is sufficient for the SSA forecast. In addition, when considering other components of the singular value decomposition, we can draw conclusions about the ratio of trending, periodicity and noise in the analyzed time series, which cannot be done using any other considered methods. ■

References

1. Mandelbrot B., Fisher A., Calvet L. (1997) A multifractal model of asset returns. *Yale Cowles Foundation for Research in Economics*, Discussion Paper No. 1164.
2. Mandelbrot B. (2004) *Fractals, case and finance*. Moscow, Izhevsk: Research Center “Regular and Chaotic Dynamics” (in Russian).
3. Peters E. (2004) *Fractal analysis of financial markets. Application of chaos in investment and economics*. Moscow: Internet Trading (in Russian).
4. Peters E. (2000) *Chaos and order in capital markets. A new analytical perspective on cycles, prices and market volatility*. Moscow: Mir (in Russian).
5. Zinenko A.V. (2012) R/S analysis in the stock market. *Business Informatics*, no. 3(21), pp. 21–27 (in Russian).
6. Cholet F. (2018) *Deep learning with Python*. Saint Petersburg: Peter (in Russian).
7. Kong Q., Han J., Jin X., Li C., Wang T., Bai Q., Chen Y. (2023) Polar motion prediction using the combination of SSA and ARMA. *Geodesy and Geodynamics*, vol. 14, no. 4, pp. 368–376. <https://doi.org/10.1016/j.geog.2022.12.004>
8. Li K., Zhang Z., Guo H., Li W., Yan Y. (2023) Prediction method of pipe joint opening-closing deformation of immersed tunnel based on singular spectrum analysis and SSA-SVR. *Applied Ocean Research*, vol. 135, 103526. <https://doi.org/10.1016/j.apor.2023.103526>
9. Montalvo C., Pantera L., Lipcsei S., Torres L.A. (2022) Signal processing applied in cortex project: From noise analysis to OMA and SSA methods. *Annals of Nuclear Energy*, vol. 175, 109193. <https://doi.org/10.1016/j.anucene.2022.109193>
10. Golyandina N.E. (2004) *Method “Caterpillar”-SSA: analysis of time series*. Saint Petersburg: St. Petersburg State University (in Russian).
11. Coussin M. (2022) Singular spectrum analysis for real-time financial cycles measurement. *Journal of International Money and Finance*, vol. 120, 102532. <https://doi.org/10.1016/j.jimonfin.2021.102532>
12. Lahmiri S., Bekiros S., Bezzina F. (2022) Evidence of the fractal market hypothesis in European industry sectors with the use of bootstrapped wavelet leaders singularity spectrum analysis. *Chaos, Solitons & Fractals*, vol. 165, part 1, 112813. <https://doi.org/10.1016/j.chaos.2022.112813>
13. Leontyeva L.N. (2011) Multidimensional caterpillar, choice of length and number of components. *Machine Learning and Data Analysis*, no. 1, pp. 5–15 (in Russian).
14. Solntsev V.N., Danilov D.L., Zhiglyavsky A.A. (1997) *Principal components of time series: Method “Caterpillar”*. Saint Petersburg: St. Petersburg State University (in Russian).
15. Baharanchi S.A., Vali M., Modares M. (2022) Noise reduction of lung sounds based on singular spectrum analysis combined with discrete cosine transform. *Applied Acoustics*, vol. 199, 109005. <https://doi.org/10.1016/j.apacoust.2022.109005>

About the author

Anna V. Zinenko

Cand. Sci. (Tech.), Associate Professor;

Associate Professor, Department of Economic and Financial Security, Siberian Federal University, 79, Svobodny Prospect, Krasnoyarsk 660041, Russia;

E-mail: anna-z@mail.ru