Neurocomputing 171 (2016) 854–865

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

# Short-term time series algebraic forecasting with mixed smoothing

Rita Palivonaite, Kristina Lukoseviciute, Minvydas Ragulskis *

*Research Group for Mathematical and Numerical Analysis of Dynamical Systems, Kaunas University of Technology, Studentu 50-147, Kaunas LT-51368, Lithuania*

ARTICLE INFO

*Article history:*
Received 28 April 2015
Received in revised form
29 June 2015
Accepted 10 July 2015
Communicated by Wei Chiang Hong
Available online 31 July 2015

*Keywords:*
Time series prediction
Smoothing
Evolutionary algorithms

ABSTRACT

Short-term time series algebraic prediction technique with mixed smoothing is presented in this paper. Evolutionary algorithms are employed for the identification of a near-optimal algebraic skeleton from the available data. Direct algebraic predictions are conciliated by internal errors of interpolation and external differences from the moving average. Computational experiments with real world time series are used to demonstrate the effectiveness of the proposed forecasting algorithm.



## 1. Introduction

Time series prediction is a challenging problem in many fields of science, engineering and finance. Time series forecasting techniques could be conditionally classified into long-term time series forecasting techniques and short-term time series forecasting techniques. In general, the object of all time series forecasting techniques is to build a model of the process and then use this model to extrapolate past behavior into future. Unfortunately, the applicability of techniques for building the model of a long time series for a short time series is often impossible simply due to the fact that the amount of available data for training, validation and testing is simply too small. On the other hand, one step-forward future horizon is often a sufficient requirement for a short-term time series predictor.

Smoothing methods can be successfully involved in short-term forecasting techniques. The use of general exponential smoothing to develop an adaptive short-term forecasting system based on the observed values of integrated hourly demand is explored in [1]. Short-term load forecasting with exponentially weighted methods is proposed in [2].

Applications of neural network techniques to short-term time series forecasting are widely used [3]. Artificial neural network (ANN) and Markov chain (MC) are used to develop a new ANN-MC model for forecasting wind speed in very short-term time scale [4]. A novel hybrid approach, combining adaptive-network-based fuzzy inference system, wavelet transform and particle swarm optimization for short-term electricity prices forecasting in a competitive market on the electricity market of mainland Spain, is presented in [5]. A similar day-based wavelet neural network method to forecast tomorrow's load is proposed in [6]. Another short-term load forecasting based on a semi-parametric additive model is presented in [7]. An adaptive neuro-fuzzy inference system to forecasting a very short-term wind time series is presented in [8]. The radial basis function ANN with a nonlinear time-varying evolution particle swarm optimization (PSO) algorithm is used to forecast one-day ahead and five-days ahead of a practical power system in [9]. PSO algorithms are employed to adjust supervised training of adaptive ANN in short-term hourly load forecasting in [10]. An adaptive two-stage hybrid network with self-organized map (SOM) and support vector machine (SVM) is used for short-term load forecasting in [11].

Artificial neural networks (ANNs) can be combined with other time series forecasting methods like autoregressive integrated moving average (ARIMA) models to take advantage of the unique strength of ARIMA and ANN models in linear and nonlinear modeling [12]. One day ahead forecasting techniques for energy demand and price prediction that combine wavelet transform (WT) with fixed and adaptive machine learning/time series models (multi-layer perceptron (MLP), radial basis functions, linear regression, or GARCH) are proposed in [13]. A hybrid methodology that combines both autoregressive integrated moving average (ARIMA) and artificial neural network (ANN) models for predicting short-term electricity prices is proposed in [14]. Short-term electricity prices hybrid forecast model that detaches high volatility and daily seasonality for electricity price based on empirical mode decomposition, seasonal adjustment and ARIMA

* Corresponding author.
*E-mail addresses:* rita.palivonaite@ktu.lt (R. Palivonaite), kristina.lukoseviciute@ktu.lt (K. Lukoseviciute), minvydas.ragulskis@ktu.lt (M. Ragulskis).

http://dx.doi.org/10.1016/j.neucom.2015.07.018
0925-2312/

is developed in [15]. The comparison of the predictions of non-linear models (artificial neural networks) with linear models (of the ARMA type) to forecast the short-term demand for electricity is presented in [16]. A robust two-step methodology for accurate wind speed forecasting based on Bayesian combination algorithm and three neural network models is proposed in [17]. A time series model of forecasting very short-term wind speed that integrates the concepts of structural breaks and Bayesian inferences, which allows prior information about the wind speeds to be incorporated into the model and boosts forecasting performance, is presented in [18].

Short-term time series forecasting techniques are successfully applied to predict weather conditions, especially wind speed. The promising results in electricity market while predicting next days electricity load demand or price implicate that short-term time series techniques can be successfully employed also in the financial market (where financial time series can be characterized by having the long-memory effect, thick tails and volatility persistence). A support vector machine (SVM) based Markov-switching multifractal (MSM) approach which exploits iterative MSM model to forecast volatility and SVM to model the innovations is proposed as a promising alternative to financial short-term volatility prediction [19]. A hybrid ANFIS based forecasting model using linear model and moving average technical index to forecast TAIEX stock is proposed in [20]. A hybrid method that combines multiple kernel learning (MKL) and a genetic algorithm (GA) to forecast short-term foreign exchange rates is presented in [21]. A three-stage nonlinear ensemble model, where three different types of neural-network based models are optimized by improved particle swarm optimization (IPSO) and generated by support vector machines (SVM) neural network, is used for a day-ahead stock e-exchange prices forecasting [22].

In spite of numerous amount of forecasting models and techniques, there cannot be a universal model that will predict everything well for all problems and there will probably not be a single best forecasting method for all situations. The combination of several prediction methods into a hybrid model could extend the limits of its applicability and could capture the patterns which are difficult to apprehend by using isolated traditional models [23]. Nevertheless, even hybrid models cannot be considered as being universal.

The main objective of this paper is to propose an algebraic short-term time series predictor with mixed smoothing which extends the functionality of algebraic predictors with external [24] and internal smoothing [25]. This paper is organized as follows. An overview of algebraic predictors with external and internal smoothing is presented in Section 2. The structure of algebraic predictor with mixed smoothing is discussed in detail in Section 3. The role of evolutionary algorithms in algebraic forecasting with mixed smoothing is presented in Section 4. Computational experiments are discussed in Section 6; concluding remarks are given in the last section.

## 2. Preliminaries

### 2.1. The rank of a sequence

An order $n$ linear recurrence sequence (LRS) with constant coefficients reads

$$x_k=\alpha_{n-1}x_{k-1}+\alpha_{n-2}x_{k-2}+\cdots+\alpha_0x_{k-n};\quad k=0,1,\ldots; \tag{1}$$

where coefficients $\alpha_j$, $j=0,1,\ldots,n-1$ are constants. The initial conditions $x_k$, $k=0,1,\ldots,n-1$ uniquely determine the evolution of this LRS [26,27]. The auxiliary polynomial to Eq. (1) reads

$$P(\rho)=\rho^n-\alpha_{n-1}\rho^{n-1}-\alpha_{n-2}\rho^{n-2}-\cdots-\alpha_0. \tag{2}$$

If the $n$ roots $\rho_1,\rho_2,\ldots,\rho_n$ of Eq. (2) are all distinct then the LRS takes the form

$$x_j=\mu_1\rho_1^j+\mu_2\rho_2^j+\cdots+\mu_n\rho_n^j; \tag{3}$$

where the coefficients $\mu_1,\mu_2,\ldots,\mu_n$ are determined in order to fit the initial conditions of the recurrence. Note that all roots are real or complex conjugate if only LRS is real. If though some roots coincide, then the recurrence reads

$$x_j=\sum_{k=1}^{r}\sum_{l=0}^{n_k-1}\mu_{kl}\binom{j}{l}\rho_k^{j-l}; \tag{4}$$

where $r$ is the number of distinct roots; $n_k$ is the multiplicity index of the $k$th root; $n_1+n_2+\cdots+n_r=n$.

The algorithm for the reconstruction of the model of LRS from a sequence $(x_j)_{j=0}^{+\infty}$ is more complex if the order of RMS is not known beforehand. Hankel transform of $(x_j)_{j=0}^{+\infty}$ yields the sequence $(h_j)_{j=0}^{+\infty}$ where $h_j=\det H_j$ and $H_j=\left(x_{k+l-2}\right)_{1\le k,l\le(j+1)}$ is a Hankel catalecticant matrix (the dimensions of $H_j$ are $(j+1)\times(j+1)$). If there exists $n\ge1$ such that $h_n\neq0$ but $h_k=0$ for all $k>n$, then $(x_j)_{j=0}^{+\infty}$ is LRS, its order is $n$ and the auxiliary equation (2) now reads

$$\begin{vmatrix} x_0 & x_1 & \cdots & x_n \\ x_1 & x_2 & \cdots & x_{n+1} \\ \cdots & \cdots & \cdots & \cdots \\ x_{n-1} & x_n & \cdots & x_{2n-1} \\ 1 & \rho & \cdots & \rho^n \end{vmatrix}=0. \tag{5}$$

This linear system of algebraic equations has one and the only one solution because $h_n\neq0$ [28].

The following example illustrates the identification of the algebraic model of a time series. Let us consider a sequence

$$S=(2,5,14,42,130,406,1266,\ldots).$$

The proposed sequence is an order $n=3$ linear recurrence sequence because $h_k=0$ for all $k>3$. The auxiliary equation

$$\begin{vmatrix} 2 & 5 & 14 & 42 \\ 5 & 14 & 42 & 130 \\ 14 & 42 & 130 & 406 \\ 1 & \rho & \rho^2 & \rho^3 \end{vmatrix}=-2\rho^3+14\rho^2-32\rho+24=0$$

yields roots $\rho_1=\rho_2=2;\rho_3=3$. Then $n_1=2;n_2=1$ and (4) takes the following form: $x_n=\mu_{10}\rho_1^n+\mu_{11}n\rho_1^{n-1}+\mu_{20}\rho_2^n$. The linear algebraic system for identification of coefficients $\mu_{10}$, $\mu_{11}$, $\mu_{20}$ reads

$$\begin{bmatrix} 1 & 0 & 1 \\ \rho_1 & 1 & \rho_2 \\ \rho_1^2 & 2\rho_1 & \rho_2^2 \end{bmatrix}\begin{bmatrix} \mu_{10} \\ \mu_{11} \\ \mu_{20} \end{bmatrix}=\begin{bmatrix} 2 \\ 5 \\ 14 \end{bmatrix}.$$

Solutions are $\mu_{10}=0$, $\mu_{11}=-1$, $\mu_{20}=2$. Thus, finally, $x_n=2\cdot3^n-n\cdot2^{n-1}$.

**Corollary 1.** *The order of a random sequence is infinite. The proof is straightforward. Assume that the order of a random sequence is finite. Then, according to Eq. (4), it is possible to reconstruct the algebraic model governing the evolution of this random sequence. Thus, the dynamics of the sequence is deterministic, what contradicts to the definition of a random sequence.*

**Corollary 2.** *Let the order of $(x_j)_{j=0}^{+\infty}$ is $m$ and the order of a random sequence $(\varepsilon_j)_{j=0}^{+\infty}$ is infinite. Then the order of $(x_j+\varepsilon_j)_{j=0}^{+\infty}$ is infinite. The proof follows from Eq. (4).*

All real world time series are inevitably contaminated with an additive noise. Thus a straightforward application of algebraic LRS theory is impossible to real world time series. Special adaptive

algebraic techniques must be developed before exact algebraic models can be employed in time series forecasting techniques.

### 2.2. Algebraic prediction, external smoothing (APES)

Algebraic polynomials of moderate order are rather efficient to approximate gradually varying one-variable functions without discontinuities or breaks [29]. However, the greater the model dimension becomes, the less is the probability of successful modeling results with algebraic polynomials. APES predictor [24] helps to solve many of the problems associated to the model dimension and the variability of the time series. The basic idea of this prediction technique can be described as follows. Let $2n+1$ observations are available for building the model of the process: $(x_k)_{k=0}^{+\infty}$; where $2n$ is the value of the observation at the current moment. It is assumed that $d_n \neq 0$ – otherwise the algebraic model of the process can be reconstructed from Eq. (5) and the prediction would be straightforward. But since that is not the case, a proposition is made that the available observations are produced from some sort of algebraic skeleton sequence by adding the external noise to its elements. In other words, the goal is to identify such vector of corrections $(\varepsilon_0, \varepsilon_1, \ldots, \varepsilon_{2n})$ that the following equality would hold true

$$\tilde{d}_n = \det\begin{bmatrix} x_0-\varepsilon_0 & x_1-\varepsilon_1 & \cdots & x_n-\varepsilon_n \\ x_1-\varepsilon_1 & x_2-\varepsilon_2 & \cdots & x_{n+1}-\varepsilon_{n+1} \\ \cdots & \cdots & \cdots & \cdots \\ x_n-\varepsilon_n & x_{n+1}-\varepsilon_{n+1} & \cdots & x_{2n}-\varepsilon_{2n} \end{bmatrix} \to 0. \quad (6)$$

Unfortunately, this is an ill-posed inverse problem and the identification of the skeleton algebraic sequence is impossible if the vector of corrections is not associated to a concrete fitness function. The fitness function proposed in [24] reads

$$F_e(\varepsilon_0, \ldots, \varepsilon_{2n}) = \frac{1}{a\left|\tilde{d}_n\right| + \sum_{k=0}^{2n} \lambda_k |\varepsilon_k|}; \quad (7)$$

where the parameter $a$ determines the penalty proportion between the magnitude of the determinant and the sum of weighted corrections; $\lambda_k$ defines the tolerance corridor for corrections $\varepsilon_k$ (note that $F_e = +\infty$ if $d_n = 0$ and no corrections are required). Evolutionary computation strategy is developed in [24] which allows the identification of a near-optimal vector of corrections by averaging 100 reconstructed algebraic skeletons for every single prediction (the prediction horizon is 1 step). Note that the prediction error metric is not present in the expression of the fitness function. Consecutive averaging of algebraic skeletons is represented in the title of the prediction technique.

### 2.3. Algebraic prediction, internal smoothing (APIS)

The alternative prediction strategy for short time series is proposed in [25]. Now it is assumed that only $2n$ observations are available: $(x_k)_{k=0}^{2n-1}$; and $x_{2n-1}$ is the value of the observation at the present moment. Then it is possible to identify the next element of the sequence $2n$ from the equation $d_n = 0$ (if only the original sequence is actually an algebraic sequence). Unfortunately, all real world time series are contaminated by some sort of noise – therefore it is clear that such straightforward computations cannot produce acceptable predictions. Thus, APIS tries to smooth the forecast before the prediction is done – instead of trying to make a straightforward projection of the algebraic model into the future. In other words, a conciliation between the variability of the skeleton algebraic sequences and the smoothness of the averaged estimates is sought in [25]

$$F_i(\varepsilon_0, \ldots, \varepsilon_{2n-1}) = \frac{1}{a\sum_{k=0}^{2n-1} \lambda_k |\varepsilon_k| + \left|\tilde{x}_{2n} - \overline{x}_{2n}\right|}; \quad a > 0; \quad (8)$$

where $\tilde{x}_{2n}$ is determined from the equality

$$\det\begin{bmatrix} x_0-\varepsilon_0 & x_1-\varepsilon_1 & \cdots & x_n-\varepsilon_n \\ x_1-\varepsilon_1 & x_2-\varepsilon_2 & \cdots & x_{n+1}-\varepsilon_{n+1} \\ \cdots & \cdots & \cdots & \cdots \\ x_n-\varepsilon_n & x_{n+1}-\varepsilon_{n+1} & \cdots & \tilde{x}_{2n} \end{bmatrix} = 0; \quad (9)$$

and $\overline{x}_{2n}$ is the smoothed moving average of $(x_k - \varepsilon_k)_{2n-1}^{k=0}$. In other words, $F_i = +\infty$ if only a straightforward algebraic prediction produces a forecast which does coincide with the smoothed moving average. Evolutionary computation strategies are developed in [25] in order to determine near-optimal smoothed skeleton algebraic sequences. The structure of algebraic predictor with mixed smoothing is discussed in detail in the next section.

## 3. Algebraic prediction, mixed smoothing (APMS)

### 3.1. Construction of the fitness function

Let $2n+1$ observations are available for building the model of the process: $(x_k)_{k=0}^{2n}$; where $x_{2n}$ is the value of the observation at the current moment. Let us introduce the fitness function for the set of corrections $(\varepsilon_k)_{k=0}^{2n}$

$$F_m(\varepsilon_0, \varepsilon_1, \ldots, \varepsilon_{2n}) = \frac{1}{\sum_{k=0}^{2n} |\varepsilon_k| + a\hat{E} + b\left|\hat{x}_{2n+1} - \overline{x}_{2n+1}\right|}; \quad a, b > 0; \quad (10)$$

where parameters $a$ and $b$ are penalty proportions between different terms in the denominator of the fitness function; $\hat{E}$ is the error computed between the observations $(x_k)_{k=0}^{2n}$ and the reconstructed algebraic skeleton; $\hat{x}_{2n+1}$ is the algebraic prediction and $\overline{x}_{2n+1}$ is the moving average: $\overline{x}_{2n+1} = \frac{1}{s}\sum_{k=0}^{s-1} x_{2n-k}$, where $s$ is the averaging window.

The principal difference between APMS and APIS (or APES) is in the introduction of the error norm into the fitness function of corrections. But this introduction is far from being straightforward. First of all (similarly to APES) corrections $(\varepsilon_k)_{k=0}^{2n}$ are used to deform the observation sequence. Note that we do not require (6) to hold true – we do not push the absolute value of the determinant of the deformed sequence to zero

$$\det\begin{bmatrix} x_0-\varepsilon_0 & x_1-\varepsilon_1 & \cdots & x_n-\varepsilon_n \\ x_1-\varepsilon_1 & x_2-\varepsilon_2 & \cdots & x_{n+1}-\varepsilon_{n+1} \\ \cdots & \cdots & \cdots & \cdots \\ x_n-\varepsilon_n & x_{n+1}-\varepsilon_{n+1} & \cdots & x_{2n}-\varepsilon_{2n} \end{bmatrix} \neq 0. \quad (11)$$

Instead, we do neglect the value $\tilde{x}_{2n} = x_{2n} - \varepsilon_{2n}$ and assume that $(\tilde{x}_k)_{k=0}^{2n-1}$ is an algebraic sequence (in principle, $\tilde{x}_{2n}$ can be computed from Eq. (9)). Now, it is possible to construct the characteristic Hankel equation for the sequence $(\tilde{x}_k)_{k=0}^{2n-1}$ (5)

$$\det\begin{bmatrix} \tilde{x}_0 & \tilde{x}_1 & \cdots & \tilde{x}_n \\ \tilde{x}_1 & \tilde{x}_2 & \cdots & \tilde{x}_{n+1} \\ \cdots & \cdots & \cdots & \cdots \\ 1 & \rho & \cdots & \rho^n \end{bmatrix} = 0; \quad (12)$$

and to reconstruct the explicit expression of the algebraic skeleton (Eq. (4)). The algebraic prediction $\hat{x}_{2n+1}$ is now considered as the extrapolation (the horizon of the extrapolation is equal to 1) of the algebraic skeleton defined by Eq. (4).

Moreover, it is now possible to assess the quality of the algebraic skeleton by the root mean squared error (RMSE)

between the skeleton and the original observation

$$\hat{E}=\sqrt{\frac{1}{2n+1}\sum_{k=0}^{2n}(\hat{x}_k-x_k)^2}. \tag{13}$$

Note that $\hat{E}\neq\sqrt{\frac{1}{2n+1}\sum_{k=0}^{2n}\varepsilon_k^2}$.

In other words, we do seek a balance between the closest algebraic skeleton and the smoothing of current observations. The ability to assess the quality of the algebraic skeleton, to minimize the distance of this skeleton from the observation data, and to reach the closeness of the algebraic prediction to the moving average exemplifies the potential of APMS.

Time series models are usually strongly non-linear and non-stationary. The question is if an algebraic predictor could be a right choice for such time series. However, APES and APIS already demonstrated that predictors based on the evolutionary identification of near-optimal algebraic skeletons (which are neither linear nor stationary) are well applicable for non-linear and non-stationary sequences. Moreover, APES has been successfully used for the segmentation of non-stationary time series [30]. Thus, APMS seems to be a credible approach for non-linear and non-stationary time series.

### 3.2. A simple numerical example

A simple numerical example is used to illustrate the APMS forecasting scheme. Let us assume that 5 data points are available ($n=2$): $x_0=1.5$; $x_1=1.3$; $x_2=2.1$; $x_3=2.9$ and $x_4=3.7$. Let us assume that the width of the observation window for the moving average algorithm is $s=2$. Then the moving average forecast is $\overline{x}_{2n+1}=\frac{x_3+x_4}{2}=3.3$.

Without losing the generality we assume that $a=b=1$. Corrections $\varepsilon_k$ are used to construct the sequence $\tilde{x}_k=x_k-\varepsilon_k$; $k=0,\ldots,4$. Then the characteristic equation reads

$$\det\begin{bmatrix}1.5-\varepsilon_0 & 1.3-\varepsilon_1 & 2.1-\varepsilon_2\\ 1.3-\varepsilon_1 & 2.1-\varepsilon_2 & 2.9-\varepsilon_3\\ 1 & \rho & \rho^2\end{bmatrix}=0;$$

the target function to be maximized reads

$$F_m(\varepsilon_0,\varepsilon_1,\varepsilon_2,\varepsilon_3,\varepsilon_4)=\frac{1}{\sum_{k=0}^{4}|\varepsilon_k|+\sqrt{\frac{1}{5}\sum_{k=0}^{4}(\hat{x}_k-x_k)^2}+|\hat{x}_5-\overline{x}_5|}; \tag{14}$$

Let us assume that $\varepsilon_0=0.1$; $\varepsilon_1=-0.1$; $\varepsilon_2=-0.1$; $\varepsilon_3=0.1$; $\varepsilon_4=0$. Then the equality

$$\det\begin{bmatrix}1.4 & 1.4 & 2.2\\ 1.4 & 2.2 & 2.8\\ 1 & \rho & \rho^2\end{bmatrix}=1.12\rho^2-0.84\rho-0.92=0$$

yields two roots $\rho_1=-0.6058$ and $\rho_2=1.3558$. Coefficients $\mu_{10}$ and $\mu_{20}$ can be determined from the relationship:

$$\begin{pmatrix}\mu_{10}\\ \mu_{20}\end{pmatrix}=\begin{pmatrix}\rho_1^3 & \rho_2^3\\ \rho_1^4 & \rho_2^4\end{pmatrix}^{-1}\cdot\begin{pmatrix}\tilde{x}_3\\ \tilde{x}_4\end{pmatrix}=\begin{pmatrix}-0.222 & 2.492\\ 0.135 & 3.379\end{pmatrix}^{-1}\cdot\begin{pmatrix}2.8\\ 3.7\end{pmatrix}=\begin{pmatrix}-0.2209\\ 1.1037\end{pmatrix}.$$

Then, direct algebraic extrapolation $\tilde{x}_n=\sum_{k=1}^{2}\mu_{k0}\rho_k^{n-0}$; $n=0,1,2,\ldots$; $k=1,2$ reads: $\hat{x}_0=0.883$, $\hat{x}_1=1.63$, $\hat{x}_2=1.948$, $\hat{x}_3=\tilde{x}_3=2.8$, $\hat{x}_4=\tilde{x}_4=3.7$.

Now, it is possible to compute the extrapolation error $\hat{E}=\sqrt{\frac{1}{5}\sum_{k=0}^{4}(\hat{x}_k-x_k)^2}=0.323$, the sum of absolute values of corrections $\sum_{k=1}^{4}|\varepsilon_k|$ and the difference $|\hat{x}_5-\overline{x}_5|=1.775$. Finally, the value of the target function reads:

$F_m(0.1,\ -0.1,\ -0.1,\ 0.1,\ 0)=0.4002$ and the forecast is $\hat{x}_5=5.075$.

It is clear that the optimization of the target function is a rather complex computational task even for such a primitive example. One could try to use deterministic strategies – for example a full sorting algorithm on a regular grid. But even a rather coarse grid $\varepsilon_k=-0.1:0.01:0.1$; $k=0,1,\ldots,4$ results into $21^5=4,084,101$ sets of corrections. Full sorting (on this regular grid) results into $F_m(-0.05,\ -0.1,\ -0.06,\ 0.1,\ -0.1)=0.4313$ and $\hat{x}_5=5.107$.

It is obvious that more demanding predictions comprising longer windows of observations can be solved exploiting only soft computing strategies.

### 3.3. APMS forecasting strategy

The prediction strategy is illustrated by the following structural algorithm.

A. The prediction strategy.
  (1) Select the prediction error $eps>0$.
  (2) Validation. Repeat until the prediction RMSE $<eps$.
    (2.1) Initialization. Select $n$ (the width of the observation window for the algebraic model becomes $2n+1$) and $s$ (the width of the moving average window); $s\leq 2n+1$. Assign first $2n+1$ elements of the time series to the Initialization Set.
    (2.2) Execute the APMS prediction algorithm (described in Part B) for at least $2n+1$ forward steps. Compute the RMSE for the available prediction data.
    (2.3) Return to (2.1) until the condition in step (2) is not satisfied.
  (3) Perform APMS prediction (as described in Part B) for the required number of time forward steps.

B. APMS one-step forward prediction algorithm.
  (1) Compute the moving average $\overline{x}_{2n+1}$ from $(x_k)_{k=2n-s+1}^{2n}$.
  (2) Use a preselected Evolutionary Algorithm to find a near-optimal set of corrections $(\varepsilon_k)_{k=0}^{2n}$:
    (2.1) Generate the population of sets of corrections.
    (2.2) For every set of corrections, compute the fitness of the set:
      (2.2.1) Compute the sum $\sum_{k=0}^{2n}|\varepsilon_k|$.
      (2.2.2) Reconstruct the algebraic sequence $\hat{x}_{k=0}^{2n+1}$ from the deformed sequence $\tilde{x}_{k=0}^{2n}=(x_k-\varepsilon_k)_{k=0}^{2n}$ according to Eqs. (12) and (4).
      (2.2.3) Compute errors of the algebraic reconstruction $\hat{E}=\sqrt{\frac{1}{2n+1}\sum_{k=0}^{2n}(\hat{x}_k-x_k)^2}$.
      (2.2.4) Find the difference between the algebraic extrapolation and the moving average $|\hat{x}_{2n+1}-\overline{x}_{2n+1}|$.
      (2.2.5) Compute the value of $F_m(\varepsilon_0,\varepsilon_1,\varepsilon_{2n})$ according to Eq. (10).
    (2.3) Iterate the Evolutionary Algorithm until a near-optimal set of corrections is found.
  (3) Reconstruct $\hat{x}_{2n+1}$ using the near-optimal set of corrections $(\varepsilon_k)_{k=0}^{2n}$.
  (4) Shift the observation window by 1 step forward and return to step (B.1).

## 4. Evolutionary algorithms for the identification of a near-optimal set of corrections

As seen from the description of the structural algorithm in Section 3, the parameters of the optimization algorithms are preset at the beginning of the validation. The objective of this section is to determine this set of parameters. We will start the

discussion on the selection of the near-optimal set of corrections $(\varepsilon_k)_{k=0}^{2n}$ by means of the Particle Swarm Optimization (PSO) algorithms since PSO can be considered as an intuitive choice for such a complex identification problem.

### 4.1. PSO algorithms for APMS

PSO is an evolutionary computation technique based on the social behavior metaphor, first introduced by Eberhart and Kennedy in 1995 [31]. Through cooperation and competition among the potential solutions, PSO often can find optima more quickly when applied to complex optimization problems. Each individual in PSO is treated as a volume-less particle (a material point) in the $D$-dimensional space. The $i$th particle is represented by its coordinates as $X_i=(x_{i1},x_{i2},\ldots,x_{iD})$; $i=1,2,\ldots,m$, where $m$ is the population's size. The previous position giving the best fitness value of the $i$th particle in its flight trajectory is recorded as $p_i=(p_{i1},p_{i2},\ldots,p_{iD})$. The index of the best particle among all particles in the population is represented by symbol $g$; the velocity of the $i$th particle is represented as $V_i=(v_{i1},v_{i2},\ldots,v_{iD})$. The particles are manipulated according to the following equations:

$$v_{id}=wv_{id}+c_1r_1(p_{id}-x_{id})+c_2r_2\left(p_{gd}-x_{id}\right);$$
$$x_{id}=x_{id}+v_{id};\ i=1,2,\ldots,D; \tag{15}$$

where $r_1$ and $r_2$ are two random variables uniformly distributed in the interval [0,1]; $c_1$ and $c_2$ are two positive acceleration constants representing weightings of the stochastic terms that pull each particle toward the particle's best and the global best; $w$ is the inertia weight balancing the global and the local search. The original recommendation for PSO parameters was $c_1=c_2=2.0$ and $w=1$ – but later convergence analysis and parameter selection studies did show that this set of parameters may lead to the divergence of particles [32]. In 1999 Clerc indicated that that the use of constriction factor may be necessary to ensure the convergence of the PSO. In [33] Shi and Eberhart noted that Clerc's recommended constriction factor is equivalent to $c_1=c_2=1.49$ and $w=0.729$ in (15). We denote this parameter set as PSO Set 1. PSO convergence and stability analysis reported by Trelea [34] do recommend that $c_1=c_2=1.7$ and $w=0.6$ – we select this parameter set as PSO Set 2. The other widely used set of parameters is recommended by Carlisle and Dozier [35] with acceleration coefficients $c_1=2.041$, $c_2=0.948$ and inertia weight $w=0.729$ – we select this parameter set as PSO Set 3. There have been no definitive recommendations in the literature regarding the swarm size in PSO. Most researchers use a swarm size of 10–60, but there are no established guidelines [36]. We fix the PSO swarm size to 50 particles – the same swarm size was used in APES and APIS computational setups. In order to test the functionality of all three sets of PSO parameters we must preselect a time series. The Dow Jones Industrial Average time series is used for that purpose; time series data range provides 1896-05-26 to 2013-08-27 monthly index observations made up of 11 US stocks [37]. We norm this series by dividing all elements by the maximum element in this sequence; RMSE errors produced in the validation step are presented in Table 1 at different orders $n$.

### 4.2. GA algorithms for APMS

Genetic algorithms are adaptive heuristic search algorithms proposed by John Holland in 1975 [38] and are based on the evolutionary ideas of natural selection and natural genetics [39]. The population of chromosomes undergoes transformations using three primary genetic operators – selection, crossover and mutation [40]. Every chromosome in our computational setup comprises a set of corrections $(\varepsilon_j)_{j=0}^{2n}$ (genes are represented as real numbers). The initial population comprises 50 chromosomes with randomly generated values of genes; values of genes are limited in the interval $[-0.2;\ 0.2]$. The fitness function associated to every chromosome is defined by Eq. (10). An even number of chromosomes are selected to the mating population from the initial population; a random roulette wheel technique is used for that purpose. We use a one-point arithmetic crossover method (the location of the point is random for every pair of chromosomes). We not only exchange genes; two new offspring chromosomes are allowed to become more or less similar to each other and to their parent chromosomes. The arithmetic crossover algorithm can be described by the following equations:

$$\begin{aligned}
\tau_{j,L}^{(k+1)}|_- &:= \beta\tau_{j,L}^{(k)}+(1-\beta)\tau_{j,R}^{(k)};\\
\tau_{j,R}^{(k+1)}|_- &:= (1-\beta)\tau_{j,L}^{(k)}+\beta\tau_{j,R}^{(k)};\\
\tau_{j,L}^{(k+1)}|_+ &:= \beta\tau_{j,L}^{(k)}+(1-\beta)\tau_{j,R}^{(k)};\\
\tau_{j,R}^{(k+1)}|_+ &:= (1-\beta)\tau_{j,L}^{(k)}+\beta\tau_{j,R}^{(k)},
\end{aligned} \tag{16}$$

where $k$ is the generation number; $\tau_{j,L}^{(k)}$ is a $j$th gene of the left parent chromosome; $\tau_{j,R}^{(k)}$ is a $j$th gene of the right parent chromosome; $\tau_{j,L}^{(k+1)}$ is the $j$th gene of the left daughter chromosome; $\tau_{j,R}^{(k+1)}$ is the $j$th gene of the right daughter chromosome; the minus sign in the subscript denotes that the $j$th gene is above the crossover point; the plus sign in the subscript denotes that the $j$th gene is below the crossover point. In any case, $j$th genes of daughter chromosomes will fit into the interval $\left[\tau_{j,L}^{(k)};\ \tau_{j,R}^{(k)}\right]$; the parameter is the weight which governs dominant individual in reproduction and it set between 0 and 1 [41]. Note that the scheme (16) becomes a classical one-point crossover algorithm at $\beta=0$ or $\beta=1$. The crossover operator is not usually applied to all pairs of chromosomes in the intermediate population [42]. A crossover coefficient $\kappa$ characterizes a probability that the crossover procedure will be executed for a pair of chromosomes in the mating population. A mutation procedure is used to prevent convergence on one local solution and helps to seek the global solution. The mutation parameter $\mu(0\le\mu<1)$ determines the intensity of the mutation process [43]. We run through genes of all chromosomes in the current generation and replace a gene selected for the mutation by a random number evenly distributed in the interval $[-0.2;\ 0.2]$.

### 4.3. The selection of parameters of GA

In general, the selection of parameters of GA is an empirical process, though some common principles are described in [44]. The following parameters of GA must be preselected: the crossover coefficient $\kappa$; the mutation parameter $\mu$; the arithmetic crossover parameter $\beta$ and the number of generations. We will use recommendations for a classical model of GA [38]. The crossover coefficient $\kappa$ will be selected from the interval [0.5; 0.9]; the mutation parameter $\mu$ – from the interval [0; 0.15]. As mentioned previously, $\beta$ we will be varied in the interval $0\le\beta<1$. There are no definitive methods for establishing how many generations an evolutionary algorithm should run for. GA may converge on good solutions after only 80 generations; we use 40 generations in our computational experiments.

Computational tests with the Dow Jones Industrial Average time series do show that GA produces lower RSME (in the validation stage) compared to PSO algorithms (Table 1). Variation of three parameters $\kappa$, $\mu$ and $\beta$ helps to identify a near-optimal set $\kappa=0.7$; $\mu=0.1$; $\beta=0.2$ for the Dow Jones Industrial Average time series (Table 1). In principle, such parameter fitting tests could be performed for every time series to be forecasted – but we fix this set of GA parameters and will use it for the prediction of all other time series.

**Table 1**
RMSE errors produced in the validation step of the Dow Jones Industrial Average time series by APMS using PSO and GA algorithms ($n$ is the order of the algebraic model; the length of the chromosome is $2n+1$ ). Best values of RMSE (and best values of GA parameters) are shown in bold; $\kappa$ is the crossover coefficient; $\mu$ is the mutation parameter; $\beta$ is the arithmetic crossover parameter.

| **PSOmodel** | $n=2$ | $n=3$ | $n=4$ | **GA**$(\kappa,\mu,\beta)$ | $n=2$ | $n=3$ | $n=4$ |
|---|---|---|---|---|---|---|---|
| PSO (*set*1) | 0.0685 | 0.0599 | 0.0783 | GA (0.7,0,0) | 0.0628 | 0.0489 | 0.0713 |
| PSO (*set*2) | 0.0729 | **0.0574** | 0.0755 | GA (0.7,0.005,0) | 0.0616 | 0.047 | 0.0453 |
| PSO (*set*3) | 0.064 | 0.0581 | 0.0654 | GA (0.7,0.01,0) | 0.0555 | 0.0458 | 0.0434 |
| **GA**$(\kappa,\mu,\beta)$ | $n=2$ | $n=3$ | $n=4$ | GA (0.7,0.02,0) | 0.0592 | 0.0453 | 0.0455 |
| GA (0.99,0.05,0) | 0.0576 | 0.0454 | 0.0452 | GA (0.7,0.03,0) | 0.0555 | 0.0454 | 0.0418 |
| GA (0.9,0.05,0) | 0.0544 | 0.0459 | 0.0469 | GA (0.7,0.04,0) | 0.0541 | 0.0454 | 0.0439 |
| GA (0.8,0.05,0) | 0.0534 | 0.0436 | 0.0456 | GA (0.7,0.05,0) | **0.0502** | 0.0459 | 0.0424 |
| GA (**0.7**,0.05,0) | **0.0507** | **0.0435** | **0.0424** | GA (0.7,0.06,0) | 0.0623 | 0.045 | 0.0433 |
| GA (0.6,0.05,0) | 0.0541 | 0.0439 | 0.0466 | GA (0.7,0.07,0) | 0.0584 | 0.0446 | 0.0412 |
| GA (0.5,0.05,0) | 0.055 | 0.0477 | 0.0468 | GA (0.7,0.08,0) | 0.0579 | **0.0433** | 0.0437 |
| **GA**$(\kappa,\mu,\beta)$ | $n=2$ | $n=3$ | $n=4$ | GA (0.7,0.09,0) | 0.0543 | 0.0442 | 0.0425 |
| GA (0.7,0.05,0) | 0.0502 | 0.0459 | 0.0424 | GA (0.7,**0.1**,0) | **0.0502** | **0.0433** | **0.0411** |
| GA (0.7,0.05,0.1) | 0.0497 | 0.0449 | 0.0413 | GA (0.7,0.11,0) | 0.0577 | 0.0464 | 0.0508 |
| GA (0.7,0.05,**0.2**) | **0.0487** | **0.0443** | **0.039** | GA (0.7,0.12,0) | 0.0537 | 0.0487 | 0.0453 |
| GA (0.7,0.05,0.3) | 0.0501 | **0.0443** | 0.0405 | GA (0.7,0.13,0) | 0.0571 | 0.0463 | 0.0468 |
| GA (0.7,0.05,0.4) | 0.0514 | 0.0463 | 0.0431 | GA (0.7,0.14,0) | 0.0596 | 0.0455 | 0.0443 |
| GA (0.7,0.05,0.5) | 0.0498 | 0.0475 | 0.442 | GA (0.7,0.15,0) | 0.054 | 0.0488 | 0.0447 |

## 5. Training and validation of the APMS model

### 5.1. Training

We will continue with Dow Jones Industrial Average time series [37] and use it for parameter estimation of the proposed mixed smoothing prediction model. The first task of the training process is to identify the order of the sequence $n$ and the optimal smoothing value $s$. The order of the sequence is fixed in the range [2, 10]; thus the width of the observation window for the algebraic model must comprise at least 21 observations ($2\cdot 10+1=21$) for the formation of the linear system of algebraic equation (5). The validation process is required to examine different choices for the sequence order $n$ and the width of the smoothing window $s$. Prediction errors (RMSE) for different $n$ and $s$ are presented in Table 2; the lowest prediction errors are produced for $n=4$ and $s=1$ (naïve moving averaging). Note that some cells in Table 2 are empty – a longer smoothing window than the algebraic window is considered as an infeasible choice.

**Table 2**
APMS training – the selection of the order of the algebraic model $n$ and the width of the smoothing window $s$ for Dow-Jones Industrial Average time series (the lowest RMSE is in bold).

| $n$ | $s=1$ | $s=2$ | $s=3$ | $s=4$ | $s=5$ | $s=10$ |
|---|---|---|---|---|---|---|
| 2 | 0.0457 | 0.0556 | 0.0608 | 0.0719 | 0.0694 | – |
| 3 | 0.0431 | 0.0517 | 0.0639 | 0.0707 | 0.780 | – |
| 4 | **0.0389** | 0.0481 | 0.0608 | 0.0728 | 0.0658 | – |
| 5 | 0.0442 | 0.0553 | 0.0647 | 0.0729 | 0.0818 | 0.1098 |
| 6 | 0.0456 | 0.0548 | 0.0640 | 0.0742 | 0.0814 | 0.112 |
| 7 | 0.0435 | 0.0563 | 0.0651 | 0.0722 | 0.762 | 0.1107 |
| 8 | 0.0438 | 0.0559 | 0.0637 | 0.0737 | 0.0827 | 0.1116 |
| 9 | 0.046 | 0.0584 | 0.0659 | 0.0703 | 0.0817 | 0.1145 |
| 10 | 0.0454 | 0.0552 | 0.0652 | 0.077 | 0.0806 | 0.1185 |

### 5.2. The adequacy of the model

Checking for adequacy of the model basically comprises the examination of the prediction residuals in the training set – residuals must not differ significantly from pure random errors ("white noise") with zero mean [45–47]. For the white noise, the theoretical auto-correlation function (ACF) and the partial auto-correlation function (PACF) are both zero at all lags. In practice, all the ACF and the PACF values of residuals should be within the confidence bounds. It is also necessary to check if the residuals are normally distributed and uncorrelated. Several diagnostic statistics and plots of the residuals can be used to examine the goodness of fit of the model to the historical data. The first of these is the Ljung–Box portmanteau $Q$ statistic. $Q$ statistic is a test of the null hypothesis that the ACF does not differ from zero, up to lag $k$ [47,48]

$$Q = n(n+2)\sum_{k=1}^{h}\frac{\hat{\rho}_k^2}{n-k}, \tag{17}$$

where $n$ is the sample size; $\hat{\rho}_k^2$ is the sample autocorrelation at lag $k$, and $h$ is the number of lags being tested. The Ljung–Box test rejects the null hypothesis (indicating that the model has significant lack of fit) if $Q>\chi^2_{1-\alpha,h}$, where $\chi^2_{1-\alpha,h}$ is the $\alpha$-quantile of the Chi-squared distribution with $h$ degrees of freedom. Lag-$h$ autocorrelation (the linear dependence of a variable with itself at two points in time) is given by

$$\hat{\rho}_h = Corr(x_t,\ x_{t-h}) = \frac{C_h}{C_0}, \tag{18}$$

where $C_h$ is the autocovariance function

$$C_h = \frac{1}{N}\sum_{t=1}^{N-h}(x_t-\overline{x})(x_{t+h}-\overline{x}). \tag{19}$$

The denominator $C_0$ is the lag 0 covariance, i.e., the unconditional variance of the process. The second test is to examine the ACF and PACF plots of the residuals [45,49]. Since the residuals should be white noise, the ACF and the PACF of the residuals should contain no hint of being predictable. Such autocorrelations and partial autocorrelations should be near zero (between the confidence limits) for any-and-all time-lag separations. If the process is not a random, then the ACF or/and the PACF are not zero. Another way to test a goodness-of-fit is the Jarque–Bera test. This is a test of null hypothesis that residuals have normal distribution, its skewness is zero and excess kurtosis is zero. The Jarque–Bera test statistic reads

$$JB = \frac{n}{6}\left(S^2+\frac{1}{4}(K-3)^2\right), \tag{20}$$

where $n$ is the number of observations (or degrees of freedom in

general); $S$ is the sample skewness, and $K$ is the sample kurtosis [50,51]. It turns out that this test statistic is comparable with a Chi-square distribution with 2 degrees of freedom. The null hypothesis of normality is rejected if the calculated test statistic exceeds a critical value from the $\chi^2_{1-\alpha,h}$.

We test the randomness of the residuals for APMS model training set. The ACF and the PCF of residuals show that there are no significantly correlated coefficients. The Ljung–Box test $Q$ statistic yields 22.9862 and does not exceed the critical value 31.4104. So this test indicates that the model has a significant lack of fit for the residuals. Since both tests did not reject that residuals are randomly distributed, we consider that model selection is adequate (Fig. 1).

The selected APMS model is used to predict the Dow–Jones Industrial Average time series – one step-forward predictions are illustrated in Fig. 2(A) and the prediction errors – in Fig. 2(B) (MAE and RMSE are used to assess the performance of the predictor).

## 6. Computational experiments

Computational experiments with real world time series are performed and assessed in this section. MA, SES, ARIMA, APES, APIS, APMS models are tested on the Dow–Jones Industrial Average time series, the Standard&Poor's index time series, St. Louis Fed Financial Stress Index (STLFSI) time series, Odonovan and Montgome time series. Results of the prediction by using different models are presented in Table 3.

### 6.1. Comparisons with the Dow–Jones industrial average time series

We continue computational experiments with the Dow–Jones Industrial Average time series and compare the functionality of the proposed forecasting technique with other methods. Errors of predictions produced by MA at $s=1$ for the Dow–Jones Industrial Average time series are illustrated in Fig. 2(C) ($s=1$ was the best selection for MA). Note that the produced RMSE value does not outperform APMS predictions. Moreover, MA at $s=1$ produces direct projection of the preceding data into the future – APMS is a much more sophisticated predictor in that sense.

The next comparison is performed with Simple Exponential Smoothing (SES) method. SES assigns exponentially decreasing weights as the observation get older

$$S_t = \alpha x_{t-1} + (1-\alpha) S_{t-1}, \tag{21}$$

where $t=1, 2, 3, \ldots$; $S_1 = x_0$ and $\alpha$ is smoothing factor; $0 \le \alpha \le 1$ [52,53]. A series of computational experiments is performed for the identification of the best value of RMSE – best results are produced at $\alpha = 0.99$ and illustrated in Fig. 2(D).

The functionality of APMS is also compared with the algebraic predictor with external smoothing (APES) [24] (results are illustrated in Fig. 2(F)) and the algebraic prediction with internal smoothing (APIS) [25]. Both APES and APIS models comprise the same order of the Hankel matrix for the Dow–Jones Industrial Average time series. It is clear that APES (at $s=1$) is not capable to produce good predictions of the Dow–Jones Industrial Average time series. APIS performs much better; prediction results are illustrated in Fig. 2(G).

Next, computational experiments are continued with Box–Jenkins's time series autoregressive integrated moving average procedure ARIMA [47] (Fig. 2(E)). First 51 values of Dow Jones industrial monthly time series are used to estimate the ARIMA model (ARIMA recommendations require to use at least 50 data points to make statistically significant forecasts).

ACF and PCF of differenced series are more consistent with a stationary process. There are no significantly correlated coefficients. Dickey–Fuller test rejects the null hypothesis of a unit root that is present in an autoregressive model. The order of the integrated part is set to 1.

Different models of ARIMA can be compared not only by analyzing ACF and PACF functions, but also information criterions, like Akaike's Information Criterion (AIC), can be used to select an appropriate model [54]. Though visual inspection of ACF and PACF provides a useful way to construct ARIMA model, the parameter selection based on criterions like AIC is more objective [55]. AIC is statistical model fit measure based on information theory. It quantifies the relative goodness of fit of various previously derived statistical models. The AIC provides a measure, which balances between two approaches: the complexity of the model and the

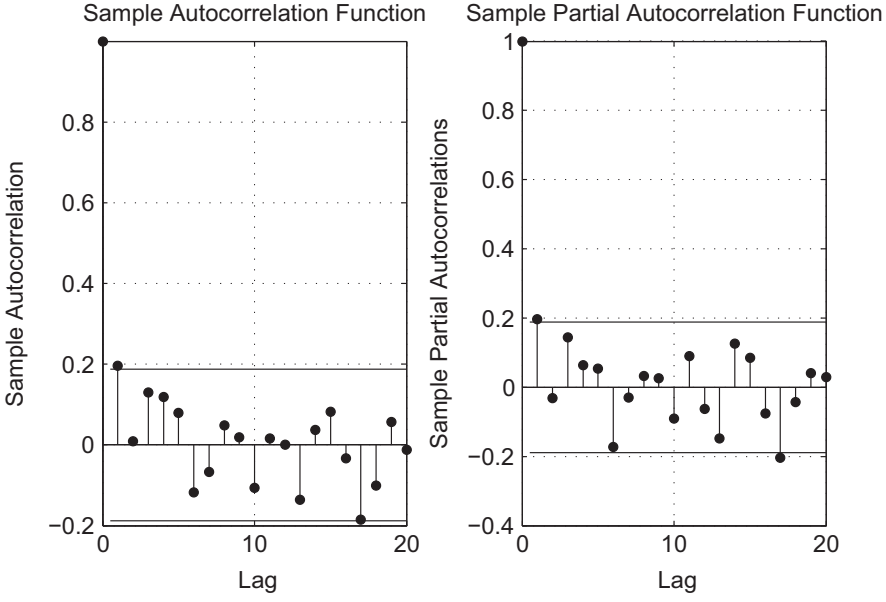


**Fig. 1.** The ACF and PACF plots of residuals of the APMS forecasting model for the first 51 points of the Dow Jones time series. The ACF and the PACF values of residuals mostly are within the confidence bounds and display no patterns; the residuals are uncorrelated and can be considered as white noise. This proves that selected APMS model is adequate.

goodness of fit of the sample data. The formula of AIC [49] is

$$AIC = 2\left(\frac{k}{n}\right) - 2\frac{\ln(L)}{n}, \tag{22}$$

where $L$ is the maximized value of the likelihood function (a function of the parameters of a statistical model), $k$ is the number of parameters in the model and $n$ is the number of observations. By comparing different forecasting models, the selection of ARIMA parameters is based on AIC criterion, though the Box and Jenkins scheme to specify and validate the model is also considered. Model validation is based on examination of the residuals if they are normally distributed and uncorrelated.

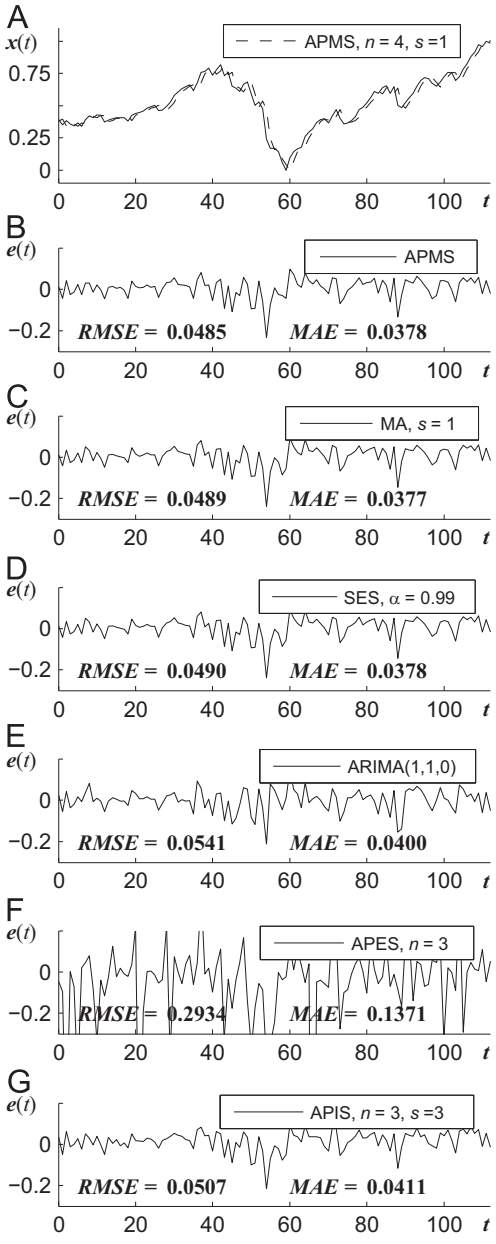


**Fig. 2.** The Dow Jones Industrial Average time series (A) and prediction errors for (B) APMS with $n=4$, $s=1$; (C) moving average forecasting, with $s=1$; (D) SES(0.99); (E) ARIMA(1,1,0); (F) APES with $n=3$; (G) APIS with $n=3$, $s=3$.

Due to previously mentioned tests, the ARIMA(1,1,0) model can be considered as a validated model and can be used for forecasting. As we can see in Fig. 2(C), the forecasting errors of ARIMA (1,1,0) are not the lowest: RMSE is 0.0541 and MAE is 0.0400. The success of APMS with The Dow Jones Industrial Average time series can be explained by a clever balance between a rigorous algebraic model, the smoothed non-linear skeleton and evolutionary corrections – as represented in the target function (10). MA cannot reveal its power for such a nonlinear and non-stationary time series – and proceeds with the naïve predictions at $s=1$ (Fig. 2(C)). The situation is almost identical with SES ($\alpha$ is 0.99). ARIMA (whatever sophisticated the training is) yields the parameter set (1,1,0) – one could not expect much from such a model (other parameter values produce only worse errors). APES exhibits over-enthusiastic variability; APIS is over-smoothing the time series. APMS is the best choice in terms of RMSE – but MA is the best in terms of MAE. This divergence can be explained by the differences in the formulations of the error metrics. Nevertheless, APMS appears to be one of the best choices (from the tested models) for the prediction of this time series.

### 6.2. S&P 500 time series forecasting

The S&P 500 is regarded as a gauge of the large cap US equities market. The index includes 500 leading companies in leading industries of the US economy, which are publicly held on either the NYSE or NASDAQ, and covers 75% of US equities. Time series contains daily 2013-01-02 to 2013-08-30 indexes [37]. APMS training yields the order of the algebraic model $n=3$ and the moving average window $s=1$. Forecasting results are compared in Fig. 3. APMS forecasting results of this time series outperform other models according to RMSE and MAE metrics Fig. 3. S&P time series is also predicted by using MA ($s=1$) and SES ($\alpha=0.9$). The best ARIMA model for this financial time series is ARIMA(0,1,1).

APMS appears as the best predictor (from the ones tested with the S&P 500 time series) both in respect of RSME and MAE. MA cannot reveal its power at $s=1$ (results at other values of $s$ are only worse); SES does not give a better result either. ARIMA training yields the parameter set (0,1,1) which does not allow to expect very good results. APES again over-estimates the variability of the time series; APIS is not able to identify the algebraic model. The S&P 500 time series is again non-linear – but not so much non-stationary compared to The Dow Jones Industrial Average time series. It seems that APMS is well suited for non-linear but more stationary time series.

**Table 3**
Results of the prediction by using different models.

| **Model** | MA | | SES | | ARIMA | | APES | | APIS | | APMS | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Time series** | RMSE | MAE | RMSE | MAE | RMSE | MAE | RMSE | MAE | RMSE | MAE | RMSE | MAE |
| Dow Jones | 0.0489 | **0.0377** | 0.0490 | 0.0378 | 0.0541 | 0.0400 | 0.2934 | 0.1371 | 0.0507 | 0.0411 | **0.0485** | 0.0378 |
| S&P 500 | 0.0486 | 0.0389 | 0.0485 | 0.0387 | 0.0494 | 0.0382 | 0.1750 | 0.1166 | 0.1793 | 0.1608 | **0.0454** | **0.0357** |
| STLFSI | **0.0286** | **0.0189** | 0.0287 | 0.0190 | 0.0333 | 0.0231 | 0.5538 | 0.2171 | 0.0349 | 0.0229 | 0.0295 | 0.0199 |
| Odonovan | 0.2148 | 0.1966 | **0.1716** | 0.1601 | 0.1831 | **0.1515** | 0.8938 | 0.5859 | 0.1933 | 0.1758 | 0.2097 | 0.1917 |
| Montgome | 0.2095 | 0.1758 | **0.1947** | **0.1708** | 0.2014 | 0.1751 | 0.4471 | 0.3520 | 0.2112 | 0.1801 | 0.2309 | 0.1987 |

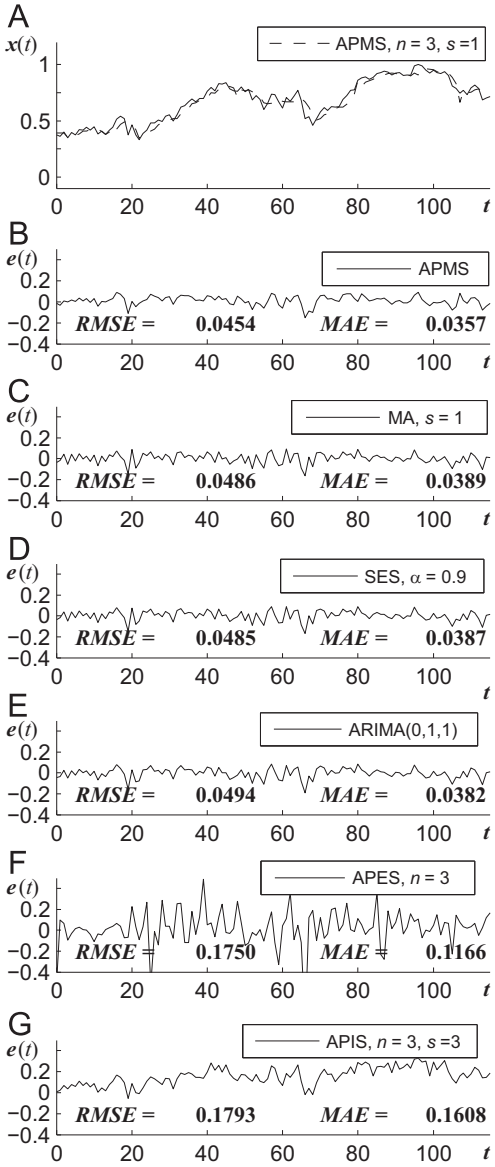


**Fig. 3.** The S&P 500 time series (A) and prediction errors for (B) APMS with $n=3$, $s=1$; (C) moving average forecasting, with $s=1$; (D) SES(0.9); (E) ARIMA(0,1,1); (F) APES with $n=3$; (G) APIS with $n=3$, $s=3$.

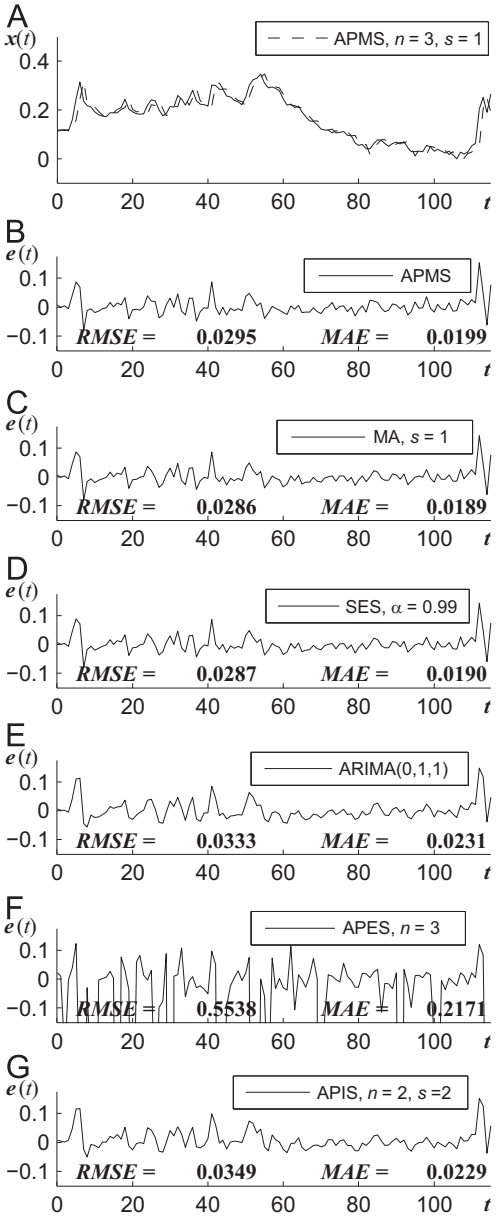


**Fig. 4.** The STLFSI time series (A) and prediction errors for (B) APMS with $n=3$, $s=1$; (C) moving average forecasting, with $s=1$; (D) SES(0.99); (E) ARIMA(0,1,1); (F) APES with $n=3$; (G) APIS with $n=2$, $s=2$.

### 6.3. STLFSI time series forecasting

The STLFSI (St. Louis Fed Financial Stress Index) measures the degree of financial stress in the markets and is constructed from 18 weekly data series: seven interest rate series, six yield spreads and five other indicators. Each of these variables captures some aspect of financial stress. Accordingly, as the level of financial stress in the economy changes, the data series are likely to move together. Data range: monthly 1993-12-31 to 2013-08-23. We normed data in the range [0, 1].

The training process is performed firstly; it yields the lowest RMSE at $n=3$ and $s=1$. Note that APMS prediction errors do not outperform MA (with window $s=1$, Fig. 4(C)) and SES ($\alpha=0.99$ , Fig. 4(B)) models. It can be noted that APIS model with parameters $n=2$, $s=2$ (Fig. 4(G)) works better than ARIMA(0,1,1) model in Fig. 4(E). The worst results, again, are obtained with APES model.

The STLFSI time series is a difficult test for all predictors – probably due to the structure of the data sequence. It seems like two different time series were merged together – a highly variable part between 0 and 55 – and a non-stationary part with a negative trend and a large end-peak from 56 to 110 (Fig. 4). Probably, the best forecasts for such a time series could be produced by a naïve predictor. However, APMS clearly outperforms ARIMA (and also APES and APIS). That demonstrates the ability of APMS to quickly adapt to changing conditions (at least quicker than ARIMA, APES and APIS).

### 6.4. Odonovan time series forecasting

Odonovan1.dat time series [56] represents consecutive yields of batch chemical processes [56]. All 70 elements of this series are normed into interval [0; 1]. Training set yields minimal RMSE when APMS parameters are set to $n=6$, $s=2$. But now APMS is not as good as compared to APIS or ARIMA models. Forecasting results with ARIMA(1;0;0) in Fig. 5(D) produced the lowest errors – RMSE=0.1831; APIS($n=2$; $s=2$) – RMSE=0.1933.

APMS is not a best predictor for the Odonovan time series – APMS is simply over-smoothing the time series (as well as MA at

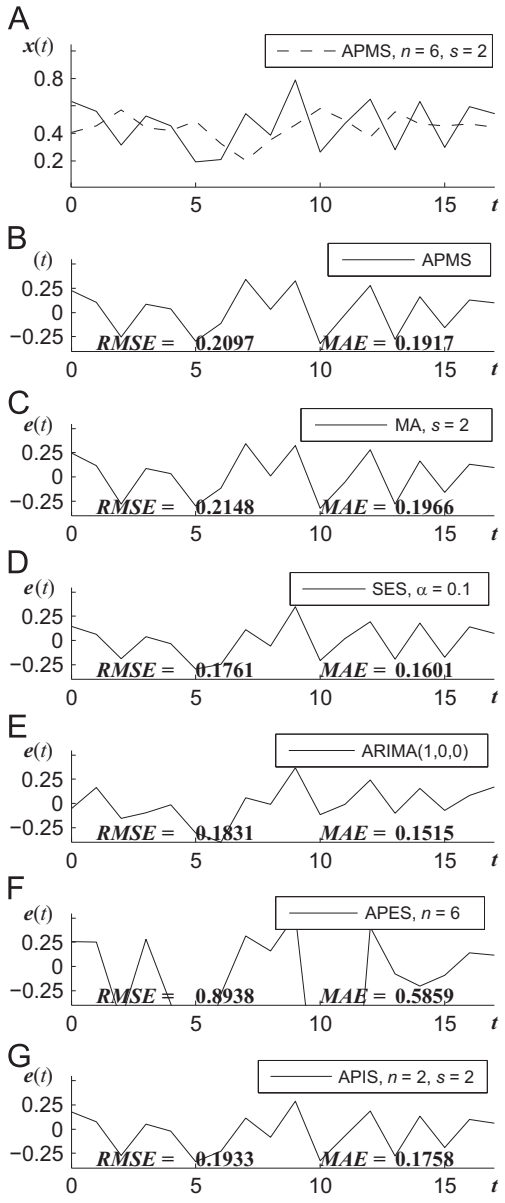


**Fig. 5.** The Odonovan time series (A) and prediction errors for (B) APMS with $n=6$, $s=2$; (C) moving average forecasting, with $s=2$; (D) SES(0.1); (E) ARIMA(1,0,0); (F) APES with $n=6$; (G) APIS with $n=2$, $s=2$.

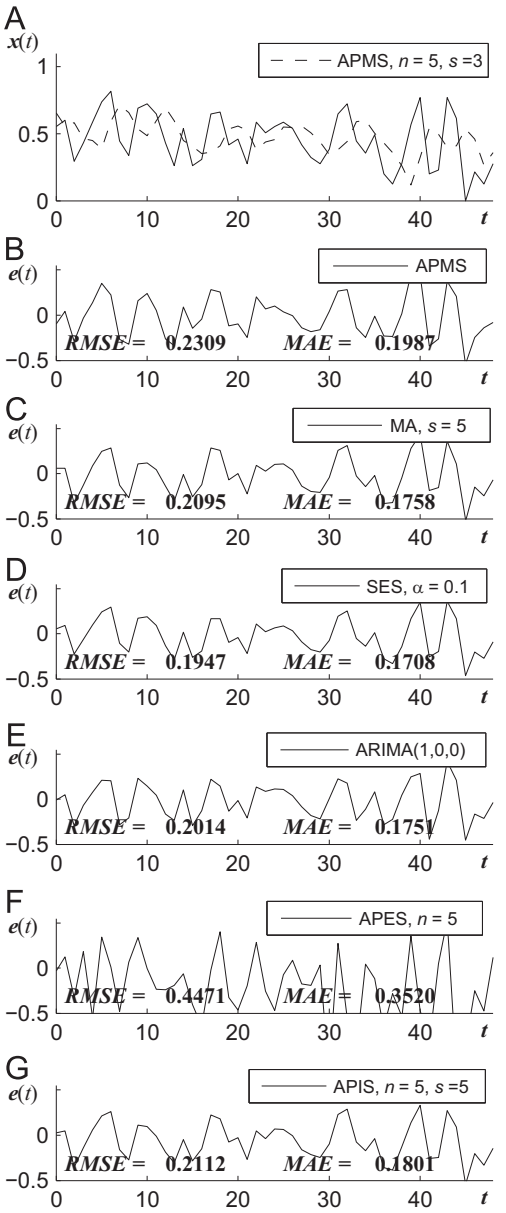


**Fig. 6.** The Montgome time series (A) and prediction errors for (B) APMS with $n=5$, $s=3$; (C) moving average forecasting, with $s=5$; (D) SES(0.1); (E) ARIMA(1,0,0); (F) APES with $n=5$; (G) APIS with $n=5$, $s=5$.

$s=2$). The best predictor from the three algebraic ones (APES, APIS and APMS) appears to be APIS. This is probably because the target function of APIS does not contain any error metrics. Therefore APIS is able to predict very short time series with minimal training what becomes important for the Odonovan time series.

### 6.5. *Montgome time series forecasting*

Montgome8.dat time series [56] comprises 100 positive elements which are normed into interval [0; 1]. APMS training set yields the lowest RMSE value at $n=5$ and $s=3$. Predictions of this model are rather variable (Fig. 6(A) and (B)), but still less than that produced by APES model (Fig. 6(F)). APIS model ($n=5$ and $s=5$) outperforms APMS and ARIMA(1,0,0) models. Best forecasting results are produced by SES model at $\alpha=0.1$.

The best predictor for the Montgome time series is SES. Again, APMS and MA are over-smoothing. And again, APIS is the best from the algebraic predictors.

## 7. Concluding remarks

APMS predictor is proposed in this paper. Evolutionary algebraic reconstruction of a near-optimal algebraic skeleton from the available data has been also exploited in APES and APIS predictors, yet APMS is a much more sophisticated predictor compared to the two ones. The first major difference of APMS is that the error metric is incorporated into the fitness function. Secondly, APMS exploits the internal and the external smoothing as integral components of the evolutionary procedure for the reconstruction of the algebraic skeleton. Finally, both APES and APIS do require a number of trials before a one-step forward prediction can be determined. APMS can produce a single prediction in one trial (pending successful training and validation). Thus, APMS is a much more computationally effective scheme and does have a definite potential for the real time forecasting of time series flows.

Computational experiments do demonstrate a good potential of APMS for financial time series forecasting results for Dow–Jones,

S&P 500, STLFSI time series are promising. However, APMS did not produce best results for Montgome and Odonovan time series. One of the possible reasons is that the pre-selected order of the algebraic model was too high, and the inherent smoothing of the moving average became the dominating feature of the predictor. In general, the main advantage of APMS over MA and SES is in the ability to reconstruct the inherent variability of the time series without a straightforward smoothing (and the associated time delay effect) when the width of the observation window becomes large. APMS offers a rather simple procedure for the selection of the predictor parameters (if compared to ARIMA) and can operate using very small data sets. The major innovation of APMS is in the identification of near optimal algebraic skeletons, which represent a conciliation between algebraic interpolants, smoothed averages and best fits in terms of the model accuracy. It seems that APMS is a good choice for non-linear but stationary time series. This is not a very unexpected result since the order of the algebraic model is not changing during the time. This feature opens possibilities for an effective exploitation of APMS in time series segmentation applications. On the other hand, APMS does not work well with very short time series. That can be explained by the fact that the error metrics is present in the target function of the APMS model, and APMS needs more data to train compared to APES or APIS models. Anyway, we can only repeat the statement mentioned in the introduction – a predictor that can outperform all other predictors in all possible situations does not (and probably will not ever) exist. Hybrid prediction models widen the areas of applicability of the prediction techniques. Thus, an incorporation of APMS into a hybrid model is a definite objective of future research.

The performance of the APMS still could be improved by improving the performance of evolutionary algorithms, or even developing a new evolutionary strategy for the identification of the near-optimal algebraic skeleton. These challenges also remain a definite objective of future research.

## Acknowledgments

Financial support from the Lithuanian Science Council under Project no. MIP-078/2015 is acknowledged.

## References

[1] W.R. Christiaanse, Short term load forecasting using general exponential smoothing, IEEE Trans. Power Appar. Syst. 90 (1971) 900–911.
[2] J.W. Taylor, Short-term load forecasting with exponentially weighted methods, IEEE Trans. Power Syst. 27 (1) (2012) 458–464.
[3] H.S. Hippert, C.E. Pedreira, R.C. Souza, Neural networks for short-term load forecasting: a review and evaluation, IEEE Trans. Power Syst. 16 (1) (2001) 44–55.
[4] S.A.P. Kani, M.M. Ardehali, Very short-term wind speed prediction: a new artificial neural network-Markov chain model, Energy Convers. Manag. 52 (1) (2011) 738–745.
[5] J.P.S. Catalao, H.M.I. Pousinho, V.M.F. Mendes, Hybrid wavelet-PSO-ANFIS approach for short-term electricity prices forecasting, IEEE Trans. Power Syst. 26 (1) (2011) 137–144.
[6] Y. Chen, P.B. Luh, C. Guan, Y. Zhao, L.D. Michel, M.A. Coolbeth, et al., Short-term load forecasting: similar day-based wavelet neural networks, IEEE Trans. Power Syst. 25 (1) (2010) 322–330.
[7] S. Fan, J.R. Hyndman, Short-term load forecasting based on a semi-parametric additive model, IEEE Trans. Power Syst. 27 (1) (2012) 134–141.
[8] C.W. Potter, M. Negnevitsky, Very short-term wind forecasting for tasmanian power generation, IEEE Trans. Power Syst. 21 (2) (2006) 965–972.
[9] C.M. Lee, C.N. Ko, Time series prediction using RBF neural networks with a nonlinear time-varying evolution PSO algorithm, Neurocomputing 73 (13) (2009) 449–460.
[10] Z.A. Bashir, M.E. El-Hawary, Applying wavelets to short-term load forecasting using PSO-based neural networks, IEEE Trans. Power Syst. 24 (1) (2009) 20–27.
[11] S. Fan, L.N. Chen, Short-term load forecasting based on an adaptive hybrid method, IEEE Trans. Power Syst. 21 (1) (2006) 392–401.
[12] G.P. Zhang, G.M. Qi, Neural network forecasting for seasonal and trend time series, Eur. J. Oper. Res. 160 (2005) 501–514.
[13] H.T. Nguyen, I.T. Nabney, Short-term electricity demand and gas price forecasts using wavelet transforms and adaptive models, Energy 35 (9) (2010) 3674–3685.
[14] P. Areekul, T. Senjyu, H. Toyama, A. Yona, A hybrid ARIMA and neural network model for short-term price forecasting in deregulated market, IEEE Trans. Power Syst. 25 (1) (2010) 524–530.
[15] Y. Dong, J.Z. Wang, H. Jiang, J. Wu, Short-term electricity price forecast based on the improved hybrid model, Energy Convers. Manag. 52 (8–9) (2011) 2987–2995.
[16] G.A. Darbellay, M. Slama, Forecasting the short-term demand for electricity—do neural networks stand a better chance? Int. J. Forecast. 16(1) (2000) 71–83.
[17] G. Li, J. Shi, J.Y. Zhou, Bayesian adaptive combination of short-term wind speed forecasts from neural network models, Renew. Energy 36 (1) (2011) 352–359.
[18] Y. Jiang, Z. Song, A. Kusiak, Very short-term wind speed forecasting with Bayesian structural break model, Renew. Energy 50 (2013) 637–647.
[19] B.H. Wang, H.J. Huang, X.L. Wang, A support vector machine based MSM model for financial short-term volatility forecasting, Neural Comput. Appl. 22 (1) (2013) 21–28.
[20] L.Y. Wei, C.H. Cheng, H.H. Wu, A hybrid ANFIS based on n-period moving average model to forecast TAIEX stock, Appl. Soft Comput. 19 (2014) 86–92.
[21] S.K. Deng, K. Yoshiyama, T. Mitsubuchi, A. Sakurai, Hybrid method of multiple kernel learning and genetic algorithm for forecasting short-term foreign exchange rates, Comput. Econ. 45 (1) (2015) 49–89.
[22] Y. Xiao, J. Xiao, F.B. Lu, S.Y. Wang, Ensemble ANNs-PSO-GA approach for day-ahead stock E-exchange prices forecasting, Int. J. Comput. Intell. Syst. 6 (1) (2013) 96–114.
[23] A.M. Rather, A. Agarwal, V.N. Sastry, Recurrent neural network and a hybrid model for prediction of stock returns, Expert Syst. Appl. 42 (6) (2015) 3232–3241.
[24] M. Ragulskis, K. Lukoseviciute, Z. Navickas, R. Palivonaite, Short-term time series forecasting based on the identification of skeleton algebraic sequences, Neurocomputing 74 (2011) 1735–1747.
[25] R. Palivonaite, M. Ragulskis, Short-term time series algebraic forecasting with internal smoothing, Neurocomputing 127 (2014) 161–171.
[26] A.V.M.V.L. Kurakin, A.S. Kuzmin, A.A. Nechavev, Linear complexity of polinear sequences, J. Math. Sci. 76 (1995) 2793–2915.
[27] H. Park, L. Eden, Matrix Rank Reduction for Data Analysis and Feature Extraction, Technical Report, 2003.
[28] Z. Navickas, L. Bikulciene, Expressions of solutions of ordinary differential equations by standard functions, Math. Model. Anal. 11 (2006) 399–412.
[29] B.B. Bezruchko, D.A. Smimov, Extracting Knowledge from Time Series: An Introduction to Nonlinear Empirical Modeling, Springer-Verlag, Berlin (2010) 410.
[30] R. Palivonaite, K. Lukoseviciute, M. Ragulskis, Algebraic segmentation of short nonstationary time series based on evolutionary prediction algorithms, Neurocomputing 121 (2013) 354–364.
[31] R.C. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Proceedings of Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, IEEE Service Center, Piscataway, NJ, 1995, pp. 39–43.
[32] F. van den Bergh, A.P. Engelbrecht, A study of particle swarm optimization particle trajectories, Inf. Sci. 176 (8) (2006) 937–971.
[33] Y.H. Shi, R.C. Eberhart, Comparing inertia weight and constriction factors in particle swarm optimization, in: Proceedings of IEEE Congress on Evolutionary Computation, San Diego, CA, IEEE Service Center, Piscataway, NJ, 2000, pp. 84–88.
[34] I.C. Trelea, The particle swarm optimization algorithm: convergence analysis and parameter selection, Inf. Process. Lett. 85 (6) (2003) 317–325.
[35] A. Carlisle, G. Dozier, An off-the-shelf PSO, in: Proceedings of the Workshop on Particle Swarm Optimization, Indianapolis, USA, 2001.
[36] J. Kennedy, R.C. Eberhart, Y.H. Shi, Swarm Intelligence, Morgan Kaufman, San Francisco, CA, 2001.
[37] Federal Reserve Bank of St. Louis, ⟨http://research.stlouisfed.org/fred2/series/STLFSI/downloaddata⟩.
[38] J.H. Holland, Adaptation in Natural and Artificial Systems: an Introductory Analysis with Application to Biology, Control, and Artificial Intelligence, The MIT Press, Cambridge, MA, 1992.
[39] D. E. Goldberg, Genetic Algorithms in Search, Optimisation, and Machine Learning, Addison Wesley Longman, Inc., Boston, MA, 1989, ISBN 0-201-15767-5.
[40] R. Kumar, Jyotishree, Blending Roulette wheel selection & rank selection in genetic algorithms, Int. J. Mach. Learn. Comput. 2 (4)(2012).
[41] O. Koksoy, T. Yalcinoz, Robust design using pareto type optimization: a genetic algorithm with arithmetic crossover, Comput. Ind. Eng. 55 (2008) 208–218.
[42] F. Herrere, M. Lozano, J.L. Verdegay, Tackling real-coded genetic algorithms: operators and tools for behavioural analysis, Artif. Intell. Rev. 12 (4) (1998) 265–319.
[43] R.L. Haupt, S.E. Haupt, Practical Genetic Algorithms, John Wiley & Sons, Inc., Hoboken, NJ, 2004.
[44] D. Whitley, A genetic algorithm tutorial, Stat. Comput. 4 (2) (1994) 65–85.
[45] G.E.P. Box, G. Jenkins, Time Series Analysis: Forecasting and Control, Holden-Day, San Francisco, CA, 1976.
[46] C. Chatfield, The Analysis of Time Series: An Introduction, fourth ed., Chapman & Hall, New York, NY, 1989.
[47] G.E.P. Box, G.M. Jenkins, G.C. Reinsel, Time Series Analysis, Forecasting and Control, 3rd ed. Prentice-Hall, Englewood Clifs, NJ, 1994.

[48] G.M. Ljung, G.E.P. Box, On a measure of lack of fit in time series models, Biometrika 65 (1978) 297–303.
[49] Peter J. Brockwell, Richard A. Davis, Introduction to Time Series and Forecasting, 2nd. ed., Springer-Verlag, New York, NY, 2002.
[50] C.M. Jarque, A.K. Bera, A test for normality of observations and regression residuals, Int. Stat. Rev. 55 (2) (1987) 163–172.
[51] P. Deb, M. Sefton, The distribution of a lagrange multiplier test of normality, Econ. Lett. 51 (1996) 123–130.
[52] ⟨http://www.itl.nist.gov/div898/handbook/pmc/section4/pmc42.htm⟩.
[53] E.S. Gardner, Jr., Exponential Smoothing: The State of the Art, Part II, ⟨http://www.bauer.uh.edu/gardner/docs/pdf/Exponential-Smoothing.pdf⟩, 2005.
[54] H. Akaike, A new look at the statistical model identification, IEEE Trans. Autom. Control 19 (1974) 716–723.
[55] U. Kumar, V.K. Jain, ARIMA forecasting of ambient air pollutants (O-3, NO, NO2 and CO), Stoch. Environ. Res. Risk Assess. 24 (5) (2010) 751–760.
[56] R.J. Hyndman, Time Series Data Library, ⟨http://datamarket.com/data/list/?⟩ q=provider:tsdl.

**Rita Palivonaite** received the M.Sc. degree in mathematics in 2008 and Ph.D. degree in Informatics in 2015 from the University of Technology, Lithuania. She is currently a lecturer within the Department of Mathematical Modeling, Kaunas University of Technology. Her current area of research interest is time series forecasting.

**Kristina Lukoseviciute** received the M.Sc. degree in mathematics in 2006 and Ph.D. degree in Informatics in 2012 from the University of Technology, Lithuania. She is currently an associate professor within the Department of Mathematical Modeling, Kaunas University of Technology. Her current areas of research interest are neuro-fuzzy systems.

**Minvydas Ragulskis** received the Ph.D. degree in 1992 from Kaunas University of Technology, Lithuania. Since 2002 he is a full professor at the Department of Mathematical Modeling, Kaunas University of Technology. His research interests include neurocomputing, nonlinear dynamical systems and numerical analysis. He is the founder and head of the Research Group for Mathematical and Numerical Analysis of Dynamical Systems (http://www.personalas.ktu.lt/~mragul).