Print ISSN: 1312-2622; Online ISSN: 2367-5357
DOI: 10.1515/itc-2018-0017

Trend and Seasonality Removal with Differential Evolution

R. Ketipov, K. Kolev, J. Sevova, I. Blagoev, P. Petrov, G. Kostadinov, I. Zankinski

Key Words: *Time series forecasting; differential evolution; information reduction.*

Abstract. *Time series forecasting is one of the high researched fields. Accurate forecasts influence many daily activities of the people and the businesses. Many difficult decisions are taken augmented by particular mathematical models introduced by time series forecasting researches. With data organized as time series many preprocessing calculations can be done before this data to be supplied at the input of the forecasting model. A common preprocessing manipulation is the removal of the trend from the time series. This type of calculation is done by linear regression and mathematical subtraction of the linear component from the original data. Usually in the time series there is seasonality. By calculation of the coefficients for sinusoidal harmonics seasonality also can be subtracted from the original data. In this research a differential evolution optimization is proposed in order the trend and the sinusoidal harmonics to be removed. By such transformation the forecasting complexity of the time series is decreased.*

1. Introduction

In day-to-day human activities, whether personal or corporate, the reliable forecasting of time series [1,2] is of great importance. This kind of predictions are the basis for making many important decisions at almost any moment in time. Even in the most trivial day-to-day activities, every modern person faces a meteorological forecast of the weather conditions of the region in which he lives or the region to which he will travel. A significant part of the meteorological forecast components are time series. Meteorological forecasting reaches its highest degree of importance when it comes to spacecraft launching.

In its nature, time series represent a sequence of measurements performed at different time points, most often at equal time intervals (but not necessarily). Such a measurement process is interesting when it comes to natural processes where it is clear that there is a proven pattern of repeatability. Examples of such measurements are the average daily temperature in a particular geographic location, the measurement of sunspots, measurement of the electricity consumption, water consumption, beverages consumption, foodstuffs consumption and others. Generally time series are visualized using points and lines that link them. Most common application of time series are in statistics, signal processing, image recognition, economics, financial mathematics, meteorological forecasts, earthquake prediction, electroencephalography, astronomy, communications, and in any other area that implies measuring at specified intervals.

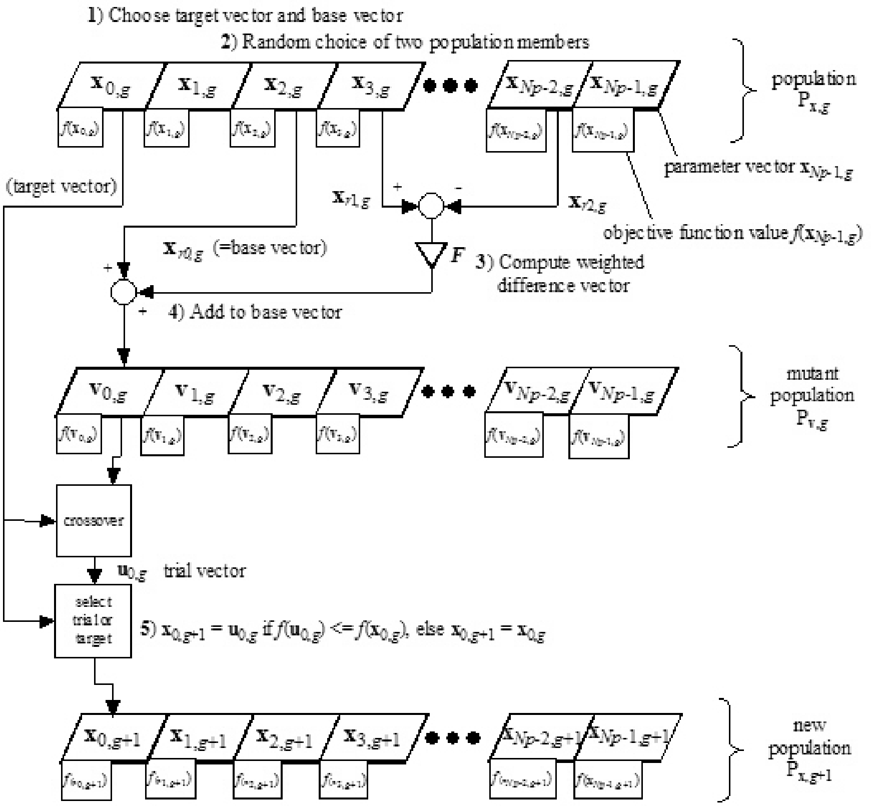
Many mathematical methods and techniques are applied in time series analysis. One of these is to decompose the time series to simpler elements. An example is the Fourier Analysis, in which information is presented in the form of sinusoidal harmonics. It is also very common for the time series to contain a linear component known more by the term “trend”. Removal of the linear component also leads to simplification of the information.

In the present study the use of global optimization heuristics called Differential Evolution to determine the coefficients describing the linear component in a time series (slope and cut in straight equation) and sinusoidal harmonic coefficients (number of harmonics, amplitude, period and phase offset) is proposed. By mathematical subtraction of the trend and the harmonics, the complexity of the input information is reduced so that the result is passed to the forecasting module. Often in practice, this forecasting module is an Artificial Neural Network.

2. Trend and Harmonics

To maximize the accuracy of the time series forecasting it is highly dependent on how precious raw data is processed before it is submitted to the forecasting module. Reducing the complexity of the data gives considerably greater possibilities for detecting dependence between individual measurements. When the forecasting module is based on a self-tuning system (for example Artificial Neural Network) better “cleaning” of the input data leads to better possibilities for dependences discover while it is also reducing the system training time.

Trend is the simplest component in a time series and removing it is one of the fastest and easiest ways to simplify the input. If considered in a slightly broader sense the trend is a kind of slope of the time series. Removing the trend transforms the measurement data into a plurality of symmetric points to rows parallel to the X axis. The mathematical trend is described by a straight line equation (1) that has two coefficients (slope and intersect). In heuristic algorithms (as it is in the Differential Evolution) the proposed solution is not always unambiguous. In terms of trend, this means there is a group of solutions that can be proposed to be subtracted from the raw data. For example, a straight of this set is the line passing through the left and rightmost measurements in the time series. Another very often used line is the value calculated after applying linear regression.



http://www1.icsi.berkeley.edu/~storn/de2.jpg

Figure 1. Differential Evolution calculation scheme

(1) $y = A * x + B.$

The coefficient A determines the slope of the line and the coefficient B determines the intersection along the Y axis when zero is given for X. In the present study, these two coefficients are determined by Differential Evolution optimization method, which may result in different values for the different algorithm starts. By their very nature, the solutions generated by Differential Evolution are suboptimal, which does not exclude and eventually fall into the global optimum.

Seasonality in the time series is due to a clearly distinguishable pattern repeatability in the process represented in the time series. This kind of repeatability can be effectively represented by decomposing the information in a set of harmonics (2).

$$
\begin{aligned}
(2)\quad y = {} & A[1] * sin(T[1]*x + W[1]) + \\
& A[2] * sin(T[2]*x + W[2]) + \\
& \ldots \\
& A[n-1] * sin(T[n-1]*x + W[n-1]) + \\
& A[n] * sin(T[n]*x + W[n]).
\end{aligned}
$$

Where n is the number of harmonics and each harmonic is characterized by – amplitudes A[i], period T[i] and phase shift W[i]. In the present study, no Fourier decomposition is used, but Differential Evolution method is used to determine the optimal number of harmonics as well as their parameters.

3. Differential Evolution for Time Series Preprocessing

Differential Evolution is a global optimization approach in the group of Genetic Algorithms. Since the method is heuristic, it does not guarantee a global optimum. In most cases, this method leads to optimal solutions close to the global optimum [1,12]. The idea for Differential Evolution creation comes from research into the effectiveness of Genetic Algorithms, which are inspired by the theory of biological evolution. In Differential Evolution method selection and crossing are applied by analogy to Genetic Algorithms, but the mutation operation affects all the elements of the individual in the population, which in Genetic Algorithms refers only to one element (*figure 1*).

In Differential Evolution method the information is presented in the form of individuals part of a population. Everyone is a vector in the solutions space. Each individual in the population is evaluated with a fitness function that determines its fitness value. Based on the individual's fitness values it is possible to choose which ones will become parents and will be recombined to create the next generation in the population. Individuals with higher fitness value have better chances of reproduction. Before the crossover operation is applied, two other individuals are selected to construct a difference vector. Difference vector applies to one of the parents and has the meaning of a mutation

Time series used for the experiments

TIME	VALUE	TIME	VALUE	TIME	VALUE	TIME	VALUE
0	24.71611971	26	47.05104128	52	48.58167901	78	47.92567409
1	27.6238817	27	47.78490496	53	47.8533154	79	47.49436844
2	29.55457013	28	49.42699883	54	47.73337138	80	47.51502127
3	30.36834623	29	51.21377468	55	48.58190838	81	48.28231028
4	30.90127713	30	51.23096599	56	48.493157	82	49.64585979
5	31.07354734	31	50.41653639	57	47.64853264	83	50.85690503
6	32.13330957	32	49.77935384	58	46.6350735	84	52.4742897
7	34.13809725	33	48.948359	59	45.21887629	85	52.64048737
8	37.15849848	34	49.49638032	60	43.04462074	86	52.17492492
9	39.83443006	35	50.31300919	61	41.46857475	87	51.74896584
10	41.10312853	36	52.63338078	62	41.40609756	88	50.05518528
11	42.81918381	37	54.22146276	63	41.63449592	89	49.41836685
12	43.68281629	38	54.89410073	64	42.42534785	90	49.81940862
13	42.42670606	39	55.35682294	65	43.39925737	91	50.40049237
14	42.29333802	40	54.69895089	66	44.35657199	92	51.93888801
15	42.27156813	41	53.74767726	67	43.97155424	93	53.22310892
16	42.67323608	42	53.32096275	68	43.14367631	94	53.32786928
17	44.21478789	43	52.1552357	69	42.37005089	95	53.32942938
18	46.05836715	44	52.78533495	70	41.58750434	96	51.94447146
19	47.14932578	45	53.8791699	71	41.25173256	97	51.2724319
20	47.8580586	46	54.45398249	72	43.20342691	98	50.24012645
21	48.12143505	47	55.22088034	73	44.23228344	99	50.57120363
22	46.74701126	48	54.85659228	74	46.4619253	100	52.07913093
23	46.07077608	49	53.36031622	75	47.98593841		
24	45.50466443	50	51.65349172	76	48.8221917		
25	45.93090059	51	49.49062036	77	48.59709404		

operation. Crossover itself is a fragment exchange operation between the two selected parents. The crossover operation gives a broader scope for exploring into the solution space, while the mutation gives fine-grained exploration into the surroundings for a particular point in the solutions space. In some cases, an elite rule is applied. It allows the most vital individuals to survive in all subsequent generations. This allows the best found solutions to survive until the evolving process is completed.

In the present study individuals in the population are encoded as a vector of real numbers (3). Since Differential Evolution is responsible for finding of the number of harmonics to be used, the size of individuals in the population can vary.

$$S[k] = \{A, B, A[1], T[1], W[i], \dots, A[s], T[s], W[s]\} \tag{3}$$

where k is the number of the individual in the population, s is the size of the vector describing the individual, A is the slope of the linear component, B is the intersect of the linear component, A[i] is the amplitude of the i-th sinus function, T[i] is the period of the i-th sinus function and W[i] is the phase shift of the i-th sinus function.

For the calculation of the fitness value an average quadratic deviation between the original time series and the calculated values of the sum of the harmonics the trend is applied. The optimization goal with Differential Evolution method is the sum of the trend and the harmonics to approximate the original data form as close as possible.

After trend and harmonics removal from the time series there is only left information that needs to be analyzed with a more sophisticated forecasting approach. The purpose of the current study is to make the data suitable for Artificial Neural Network training.

4. Experiments and Results

The experiments were performed with the time series presented numerically in the *table* and graphically in *figure 2*. The graphical representation of time series information uses a smoothing algorithm for better clarity.

The coefficients for the linear component in the time series proposed by Differential Evolution method lead to an equation of line in the following form:

$$y = 0.39 * x + 19. \tag{4}$$

Removing the linear component in the time line modifies the curve that describes it as shown in *figure 3*.

In this experiment Differential Evolution method offers five harmonics, which have the following form:

$$\begin{aligned} y = {} & 1.39 * sin(0.68 * x + 0.53) + \\ & 3.01 * sin(0.17 * x + 0.07) + \\ & 5.97 * sin(0.07 * x + 0.1) + \\ & 7.38 * sin(0.05 * x + 0.17) + \\ & 9.93 * sin(0.03 * x + 0.19). \end{aligned} \tag{5}$$

Removing values the harmonics leads to the elimination of the seasonality that is visually shown in *figure 4*.

As a result of both preprocessing steps only components that are subject to analysis with a more sophisticated forecasting approach left. Such forecasting approach are Artificial Neural Networks as described in [3-12].

Conclusion

As a result of the experiments conducted it is clear that the use of heuristics for time series preprocessing is sufficiently effective and can accelerate the training of self-adjusting forecasting systems by improving the level of predictions made, for example, with Artificial Neural Networks.

For future studies of interest it would be worthwhile to investigate the possibility of a combination with Kalman filter [13] in the preprocessing phase of the analysis. Also, the subsequent application of pre-processed time series

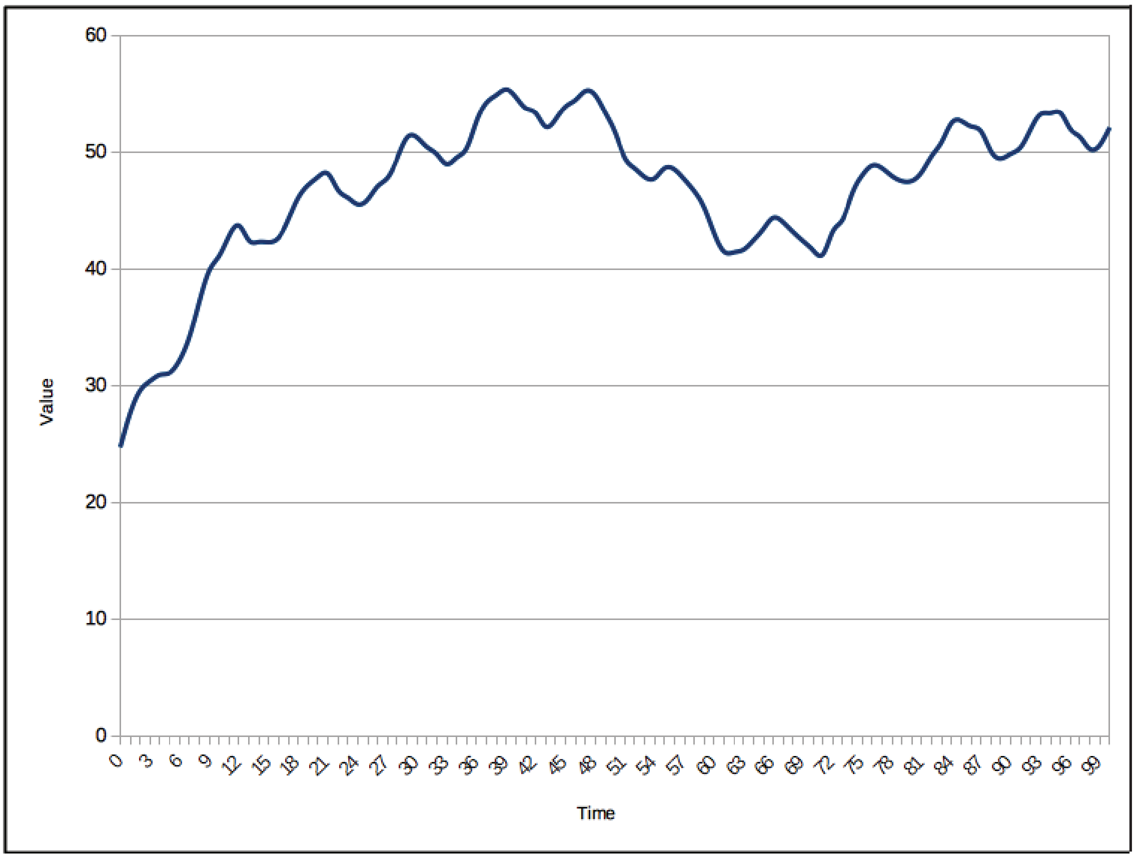


Figure 2. Visual representation of the time series used for the experiments

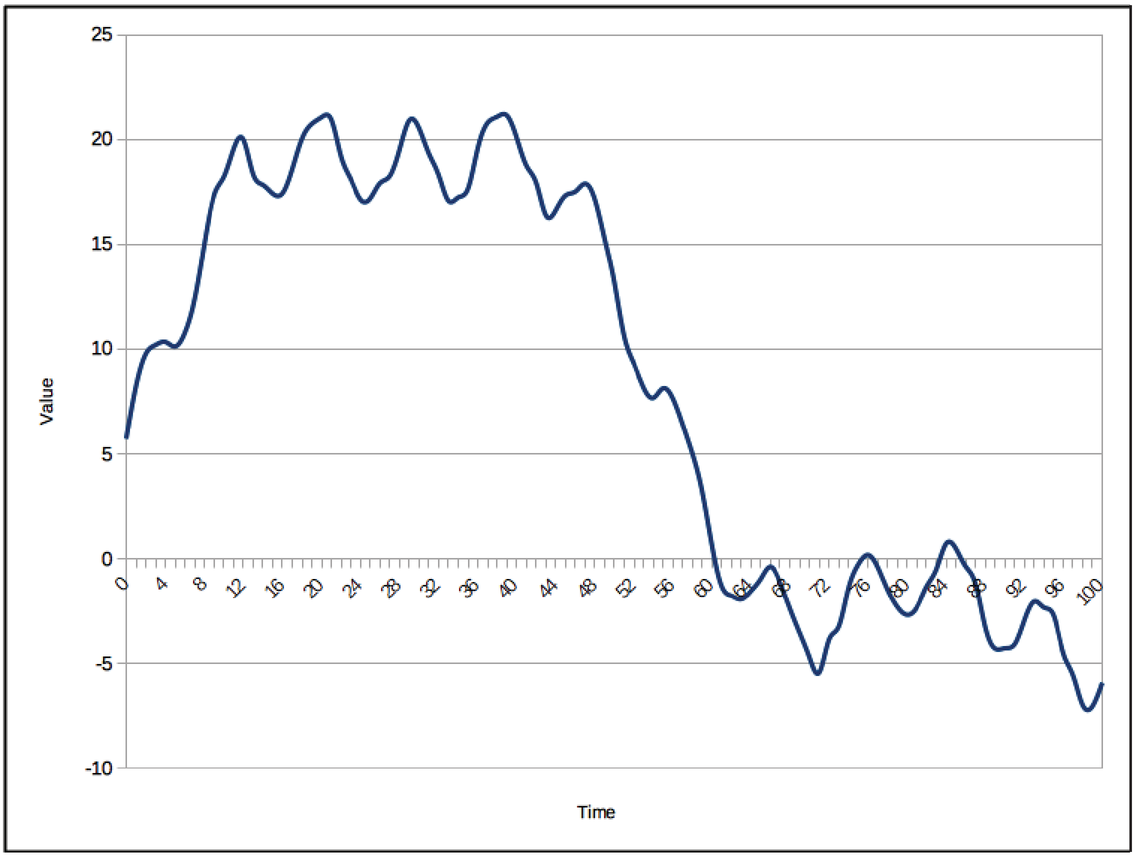


Figure 3. Time series after trend removal

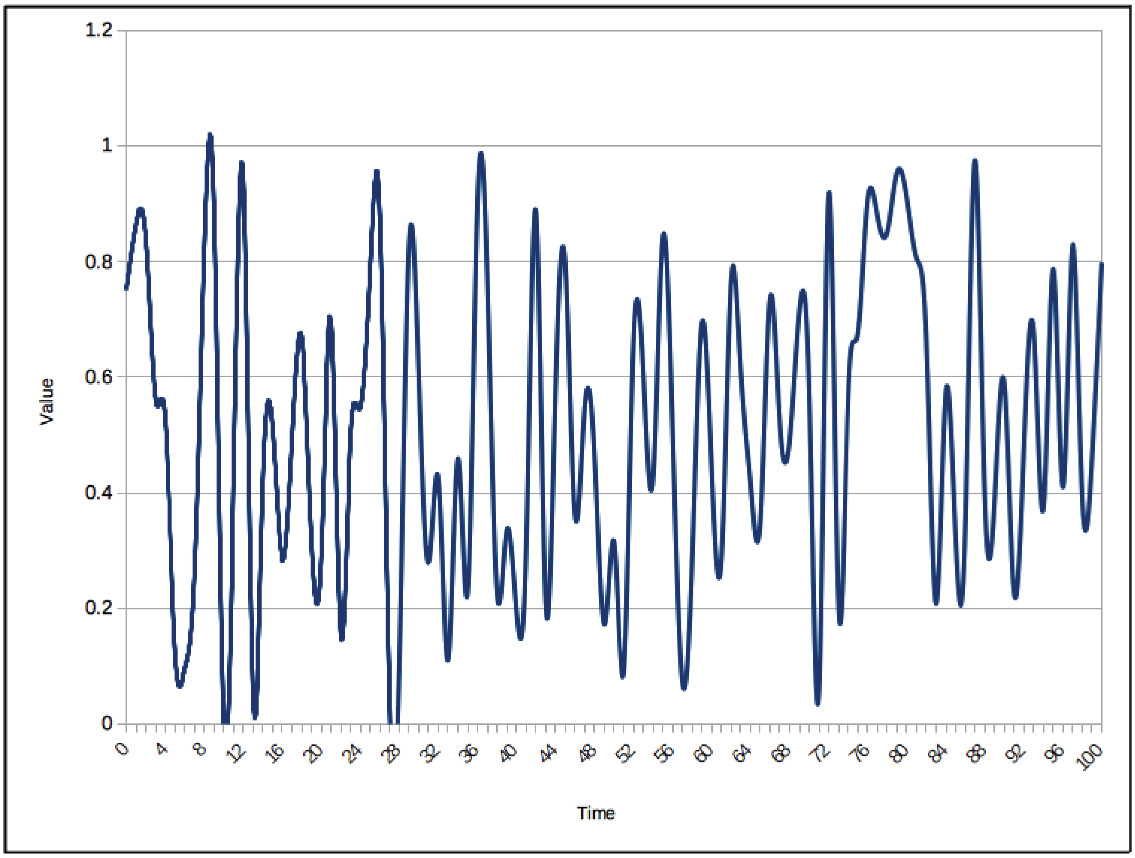


Figure 4. Time series after trend and seasonality removal

could be used to work with more non-standard Artificial Neural Networks, such as Generalized Artificial Neural Networks [14].

Acknowledgements

This work was done by ideas and private funding provided by Velbazhd Software LLC.

References

1. Balabanov, T. Avoiding Local Optimums in Distributed Population Based Heuristic Algorithms (in Bulgarian). Proceedings of XXIII International Symposium Management of Energy, Industrial and Environmental Systems, ISSN 1313-2237, Bankya, 2015, 83-86.

2. Balabanov, T. Heuristic Forecasting Approaches in Distributed Environment (in Bulgarian). Proceedings of Anniversary Scientific Conference 40 Years Department of Industrial Automation. UCTM, Sofia, 2011, 163-166.

3. Zankinski, I., T. Stoilov. Effect of the Neuron Permutation Problem on Training Artificial Neural Networks with Genetic Algorithms in Distributed Computing. Proceedings of XXIV International Symposium Management of Energy, Industrial and Environmental Systems, ISSN 1313-2237, Bankya, 2016, 53-55.

4. Balabanov, T., I. Zankinski, K. Kolev. Multilayer Perceptron Training Randomized by Second Instance of Multilayer Perceptron. Extended Abstracts of 13th Annual Meeting of the Bulgarian Section of SIAM, ISSN 1313-3357, Sofia, 2018, 16-17.

5. Balabanov, T., I. Zankinski, R. Ketipov. Weights Permutation in Multilayer Perceptron. Proceedings of International Conference on Big Data, Knowledge and Control Systems Engineering, John Atanasoff Society of Automatics and Informatics, ISSN 2367-6450, Sofia, 2018, 23-27.

6. Balabanov, T., T. Atanasova, I. Blagoev. Activation Function Permutation for Multilayer Perceptron Training. Proceedings of International Conference on Big Data, Knowledge and Control Systems Engineering, John Atanasoff Society of Automatics and Informatics, ISSN 2367-6450, Sofia, 2018, 9-14.

7. Balabanov, T., R. Ketipov, Z. Atanassova. MLP with Stochastic Manipulated Hidden Layer. Proceedings of International Scientific Conference UNITECH18, 2, ISSN 1313-230X, Gabrovo, 2018, 324-328.

8. Balabanov, T., I. Blagoev, K. Dineva. Self Rising Tri Layers MLP for Time Series Forecasting. Proceedings of International Conference on Distributed Computer and Communication Networks, ISSN 1865-0929, Moscow, 2017, 577-584.

9. Balabanov, T. Long Short Term Memory in MLP Pair. Proceedings of International Scientific Conference UniTech 2017, 2, ISSN 1313-230X, Gabrovo, 2017, 375-379.

10. Balabanov, T., K. Genova. Distributed System for Artificial Neural Networks Training Based on Mobile Devices. Proceedings of International Conference Automatics and Informatics, ISSN 1313-1850, Sofia, 2016, 49-52.

11. Balabanov, T., K. Genova. AJAX Distributed System for Evolutionary Algorithms based Artificial Neural Networks Training. Proceedings of XXIV International Symposium Management of Energy, Industrial and Environmental Systems, ISSN 1313-2237, Bankya, 2016, 49-52.

12. Balabanov, T., I. Zankinski, N. Dobrinkova. Time Series Prediction by Artificial Neural Networks and Differential Evolution in Distributed Environment. Proceedings of International Conference on Large-Scale Scientific Computing. ISBN 978-3-642-29842-4, Sozopol, 2011, 198-205.

13. Alexandrov, A. AD HOC Kalman Filter Based Fusion Algorithm for Real-time Wireless Sensor Data Integration. Proceedings of FQAS15, ISBN 978-3-319-26153-9, Heidelberg, 151-160, 2015.

14. Tashev, T., H. Radev. Modeling of the Synthesis of Information Processes with the Help of Generalized Nets – *Cybernetics and Information Technologies*, 3, 2003, No. 2, ISSN 1311-9702, 92-104.

Manuscript received on 15.04.2019

Rumen Ketipov *is a PhD student at Institute of Information and Communication Technologies – Bulgarian Academy of Sciences. His common interests are in risk analysis, management and decision making.*

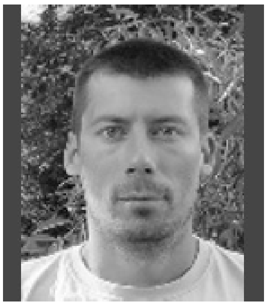
Contacts:
e-mail: rketipov@iit.bas.bg

Kolyu Kolev *is a PhD student at Institute of Information and Communication Technologies – Bulgarian Academy of Sciences. He has research interests in communications and embedded systems.*

Contacts:
e-mail: kkolev@iit.bas.bg

Janeta Sevova *is a PhD student at Institute of Information and Communication Technologies – Bulgarian Academy of Sciences. She is working in the field of industrial production lines.*

Contacts:
e-mail: jsevova@iit.bas.bg



Ivan Blagoev *is a PhD student at Institute of Information and Communication Technologies – Bulgarian Academy of Sciences. His research is in the field of financial time series forecasting.*

Contacts:
e-mail: i.blagoev@iit.bas.bg

Plamen Petrov, *is a PhD student at Institute of Information and Communication Technologies – Bulgarian Academy of Sciences. He has wide experience in education and modern learning aproaches.*

Contacts:
e-mail: p.petrov@iit.bas.bg

Georgy Kostadinov, *is a PhD student at Institute of Information and Communication Technologies – Bulgarian Academy of Sciences. He has wide experience in complex IT infrastructures and business setup.*

Contacts:
e-mail: g.kostadinov@iit.bas.bg

Iliyan Zankinski *is a programmer at Institute of Information and Communication Technologies – Bulgarian Academy of Sciences. He is working in the field of artificial intelligence applied in real life problems.*

Contacts:
e-mail: iliyan@hsi.iccs.bas.bg

Contacts:
Institute of Information and Communication Technologies
Bulgarian Academy of Sciences
Acad. Georgi Bonchev St., block 2, 1113 Sofia, Bulgaria
phone: +359 2 9793237
e-mail: iict@bas.bg