IEEE Second International Conference on Data Stream Mining & Processing
August 21-25, 2018, Lviv, Ukraine

Hybridization of the SGTM Neural-like Structure through Inputs Polynomial Extension

Pavlo Vitynskyi
Department of Publishing Information Technologies,
Lviv Polytechnic National University,
Lviv, Ukraine,
pavlo.vitynsky@gmail.com

Roman Tkachenko
Department of Publishing Information Technologies,
Lviv Polytechnic National University,
Lviv, Ukraine,
roman.tkachenko@gmail.com

Ivan Izonin
Department of Publishing Information Technologies,
Lviv Polytechnic National University,
Lviv, Ukraine,
ivanizonin@gmail.com

Hakan Kutucu
Department of Computer Engineering,
Karabuk University,
Karabuk, Turkey,
hakankutucu@karabuk.edu.tr

Abstract—**In this paper, a new approach for increasing the approximation accuracy with the use of computational intelligence tools is described. It is based on the compatible use of the neural-like structure of the Successive Geometric Transformations Model and the inputs polynomial extension. To implement such an extension, second degree Wiener polynomial is used. This combination improves the method accuracy for solving various tasks, such as classification and regression, including short-term and long-term prediction, dynamic pricing, as well as image recognition and image scaling, e-commerce. Due to the use of SGTM neural-like structure, the high speed of the system is maintained in both training and using modes. The simulation of the described approach is carried out on real data, the time results of the neural-like structure work and the accuracy results (MAPE, RMSE, R) are given. A comparison of the operation of the method with existing ones, such as Support vector regression, Classic linear SGTM neural-like structure, Linear regression (using Stochastic Gradient Descent), Random Forest, Multilayer Perceptron, AdaBoost are made. The advantages of the developed approach, in particular with regard to the highest accuracy among existing ones were experimentally established.**

Keywords—approximation, Wiener polynomial, neural-like structures, Successive Geometric Transformation Model, input's extension

I. Introduction

To solve a number of data processing tasks, such as time series predicting [1, 2]; computer network traffic modeling [3]; development of the prevention subsystems for the smart home system [4]; development of information technology for environmental monitoring [5]; medical diagnostics [6]; image processing [7, 8, 9]; dynamic pricing; there is a critical need for an accurate and fast solution of the approximation task.

One of the promising classes of methods for solving this problem are methods based on machine learning. An effective solution to the problem of data approximation, in particular on the basis of the use of ANN, is based on their ability to approximate nonlinear functions. ANN as a universal approximator is capable of reproducing dependencies of any complexity [2].

Existing methods based on machine learning provide rather good results [10]. However, the use of this tools for solving the problem is characterized by the fact that its effectiveness largely depends on the computational capabilities of a particular type of ANN, which to some extent follows from their architecture [11]. In addition, the classic neural network tools, with iterative learning algorithms, is quite slow for solving such tasks. This imposes a number of limitations, mostly on time for ANN class that can be applied while working online. Another disadvantage of a number of methods is the dimension of a training set that will provide sufficient approximation quality. In some cases, it is not enough.

All this leads to the improvement of existing and the development of new methods for increasing the accuracy of approximation in the tasks of information processing, in particular, using efficient algorithms of machine learning.

In this work, we propose a new hybrid neural-like structure of the Successive Geometric Transformations Model (SGTM NLS) for solving this task. The feature of the proposed approach is the use of Wiener polynomial for the functional extension of its inputs. This ensures high accuracy of approximation with little time spent on training.

II. Linear Neural-like Structure of Successive Geometric Transformations Model

The basis for modeling using the SGTM NLS is the basic principle of representing hypersurfaces of response in orthogonal coordinate systems (both straight-line and curvilinear) that coincide with the major dimensions of hypersurfaces.

The peculiarity of the linear-type SGTM neural-like structure is that the hypersurfaces of the response are hyperplanes [12]. In this case, the additional dimension of the model is completely determined by the noise components and round-off errors.

The training and using procedures of this computational intelligence tool are the same types [12]. Detailed training can be seen in [13]. The result of the application of the linear SGTM is that the basic measurements of the hyperplanes coincide with the results obtained using known PCA methods. However, a number of advantages characterize the method based on SGTM, in particular, it is fast due to non-iterative, without errors accumulation and noticeable dimensionality limitations. It is no need to perform iterative adaptation or to solve the systems of normal equations [14].

In usage mode of the trained model, there is a fundamental possibility of analyzing the coordinates (components of the model) for the predictability and extraction of noise components. The topology of the SGTM NLS is shown on Fig. 1.

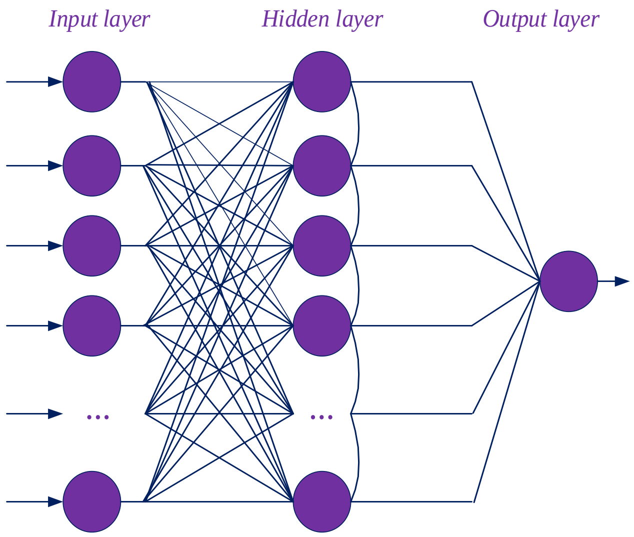


Fig. 1. The topology of the SGTM linear neural-like structure

The peculiarity of the SGTM neural-like structure is that their both variants (as software [12] as hardware [14]) effective implementation it is possible in particularly using parallel and distributed computing [15, 16].

III. Wiener Polynomial

One of the most effective methods for solving approximation tasks is based on the application of the Wiener polynomial [17]. According to the Weierstrass first theorem, this polynomial provides a simulation of continuous dependencies with arbitrarily high accuracy. Wiener's polynomial can be written as follows:

$$Y(x_1,...,x_n) = a_i + \sum_{i=1}^{n} a_i x_i + \sum_{i=1}^{n}\sum_{j=i}^{n} a_{i,j} x_i x_j + \\ + \sum_{i=1}^{n}\sum_{j=i}^{n}\sum_{l=j}^{n} a_{i,j,l} x_i x_j x_l + ... \qquad , \qquad (1) \\ ... + \sum_{i=1}^{n}\sum_{j=i}^{n}\sum_{l=j}^{n} ... \sum_{z=k-1}^{n} a_{i,j,l,...,z} x_i x_j x_l ... x_z$$

where n is number of variables and k is the degree of the Wiener polynomial.

A rather complicated problem, in this case, is the determination of the polynomial's coefficients using the various variants of the linear regression methods. It is because of the high computational complexity and the fact that this task belongs to the class of almost degenerate tasks.

The difference of this work is that we determine the polynomial's coefficients not using the least squares method, but using the hybrid variant of the SGTM NLS in supervised mode.

IV. Hybrid Neural-like Structure of Successive Geometric Transformation Model

The aim of the SGTM NLS hybridization is to provide accelerated formation of the inputs weights of neural-like structure, which will be the Wiener polynomial's coefficients and their application for solving various tasks.

The structure of the hybrid version of SGTM NLS contains two blocks (Fig. 2). The first one (Generator of the Wiener polynomial inputs) is intended to form the Wiener polynomial's members. This occurs according to (1) for a given the number of input variables (n) and the polynomial's degree (k).

Wiener polynomial's members x_i, $x_i x_j$, $x_i x_j x_l$, ... , ... , $x_i x_j x_l .... x_z$ are formed sequentially for $i = \overline{1,n}$, $j = \overline{i,n}$, $l = \overline{j,n}$, ... , $z = \overline{k-1,n}$ and $k = 1, 2, 3, \dots$.

The next block of hybrid structure is a linear version of SGTM (Fig. 1). At its inputs, we serve the members of the Wiener polynomial, formed of the primary inputs (the values of the independent variables of a specific task) and the given polynomial degree.

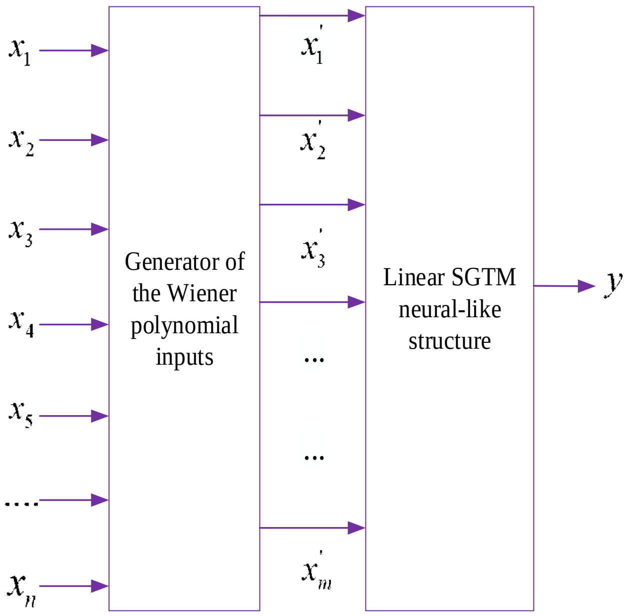


Fig. 2. Scheme of the hybrid SGTM neural-like structure

Such inputs polynomial extension of the linear version of SGTM NLS, allows increasing the accuracy of training and prediction procedures. In addition, this approach provides an opportunity to present a result in a compact form of the Wiener polynomial. This is possible given the repetition of the solution provided by the chosen learning tool. To do this, it is necessary to make a decomposition of the linear SGTM NLS [12] and synthesize the Wiener polynomial coefficients using the algorithm described in [13].

The next step in the procedure is to train SGTM NLS according to a known algorithm [12, 13], in which the neural-like structure implicitly forms the Wiener polynomial coefficients. They will be used to solve various regression problems.

V. MODELLING

The method's simulation was carried out to solve the regression task. The task was to simulate (predict) the solar radiation of Libya. Data for the task was collected in 25 cities of Libya (Fig. 3) in the period 2010-2015 [18]. The training sample has 1900 vectors, each of which contains the seven inputs and one output attributes (Table I).

TABLE I. TRAINING DATA'S ATTRIBUTES

Variable	Attributes
x_1	Month
x_2	Elevation
x_3	Mean Temperature
x_4	Relative Humidity
x_5	Mean sunshine duration/h
x_6	Longitude
x_7	Latitude
y	Daily solar radiation

The entire sample was randomly divided into proportions of 80% and 20% [19] for the implementation of training and testing procedures respectively.

Basics parameters of the hybrid SGTM neural-like structure are m inputs, m neurons in the hidden layer, 1 output, where:

$$m = n + \frac{n(n+1)}{2} \ . \qquad (2)$$

The Wiener polynomial degree that used for experiments was $k = 2$.

The quality assessment results of the developed hybrid structure were obtained using such indicators [20, 21]:

- tool absolute percentage error (MAPE) [22];

$$MAPE = \frac{1}{m}\sum_{i=1}^{m}\left|\frac{y_i - y_i^{'}}{y_i}\right| * 100 \ . \qquad (3)$$

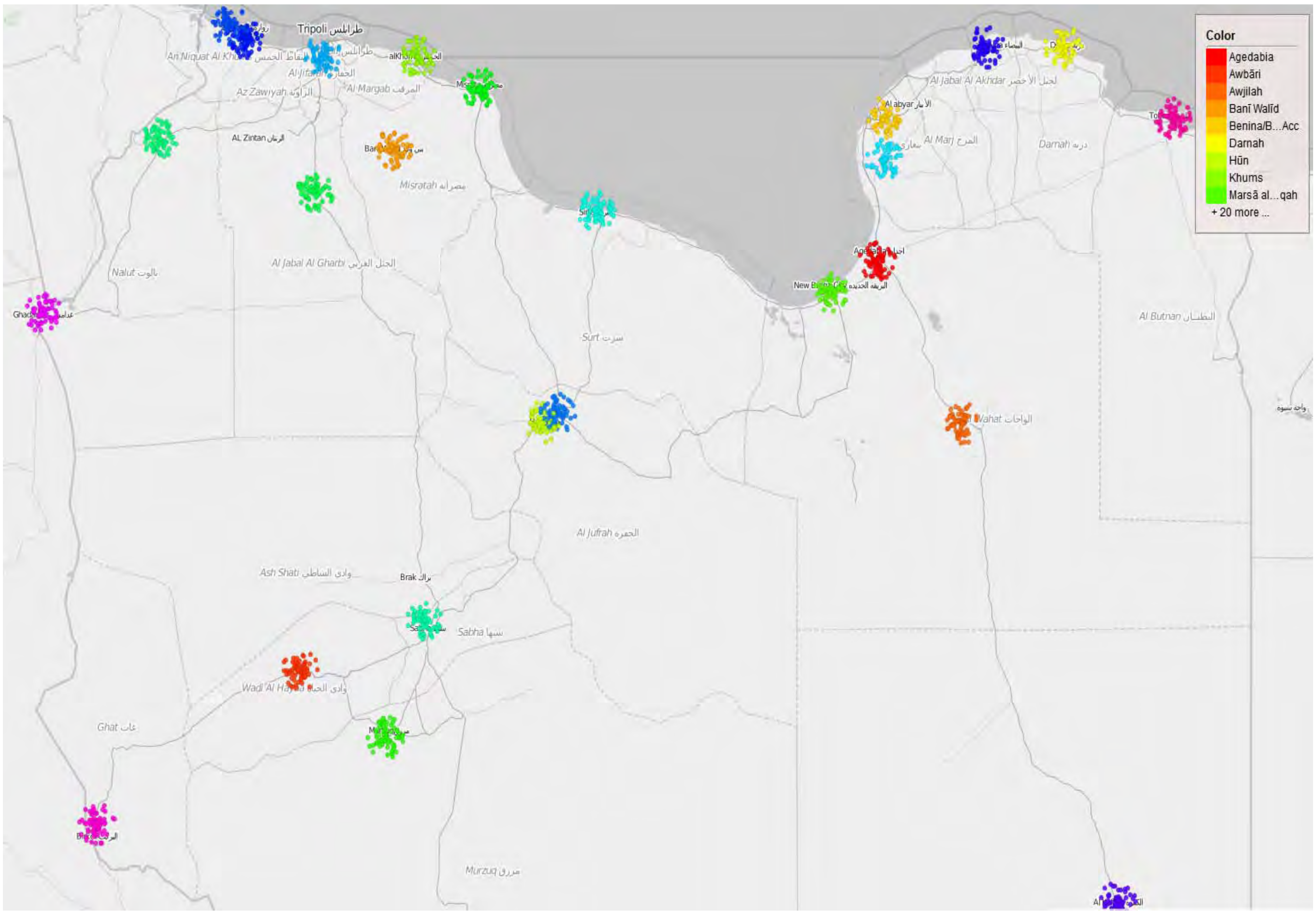


Fig. 3. Map of the Libya's cities.

- root tool squared error (RMSE) [23]:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{m}\left(y_i - y_i^{'}\right)^2}{m}} \ , \qquad (4)$$

where y_i is a real daily solar radiation value and $y_i^{'}$ is the predicted value, $i = \overline{1, m}$.

- linear correlation coefficient (R) [24]:

$$R = \frac{m(\sum xy) - (\sum x)(\sum y)}{\sqrt{\left(m\sum x^2 - (\sum x)^2\right)\left(m\sum y^2 - (\sum y)^2\right)}}. \quad (5)$$

The software solution for implementing the described approach was developed using Python. The NumPy library, which contains linear algebra operations and provides high-performance, was used for work with data arrays.

The results of the quality assessment of the described approach based on (3), (4), (5) are given in Table II. Results visualization of the described approach is shown in Fig. 6(g).

VI. Comparison of the Simulated Results

Comparison of the results of the developed approach occurred with known methods. Their parameters that were used for simulation are given in Table III.

TABLE II. MODELLING RESULTS

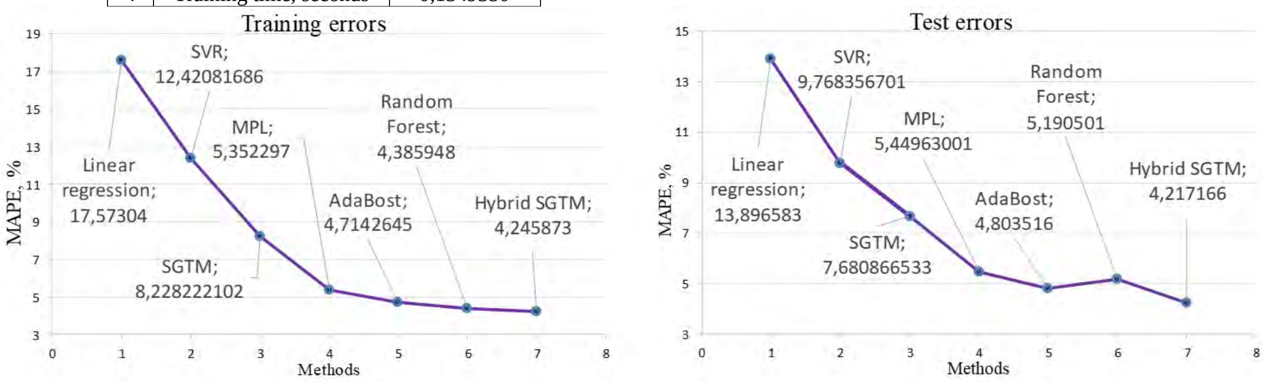
N	Indicator	Result
1	MAPE, %	4,217
2	RMSE	0,298
3	R	0,98
4	Training time, seconds	0,1349350

Implementations of all compared methods were taken from the scikit-learn machine learning library in Python.

TABLE III. EXISTING METHODS

N	Method	Parameters
1	Support vector regression	kernel='rbf', gamma='auto', coef0=0.0, epsilon=0.001, max_iter=200
2	Clasic linear SGTM neural-like structure	n inputs (primary data from Table 1), n neurons in the hidden layer, 1 output
3	Linier Regression (with Stochastic Gradient Descent)	loss = 'squared_loss', alpha=0.0001
4	RandomForest	max_depth=5, random_state=0
5	Multilayer Perceptron	hidden_layer_sizes=(100, 40, 20), activation='relu', solver='adam', alpha=0.0001, batch_size='auto', max_iter=200
6	AdaBoost	max_depth=4, n_estimators=300

In Fig. 4, the results of comparison with other methods in training and testing modes are shown. As can be seen from Fig. 4, the best results for both modes are obtained using the developed method (based on MAPE). The same results are confirmed by other indicators.



a) b)

Fig. 4. Accuracy comparison of different methods through MAPE: a) training errors: b) test errors.

The quality evaluation of the result obtained while comparing with other methods was based beside (3) and (4) on the comparison of the training procedure duration [22].

In Fig. 5, the duration of the training procedure of the all methods in seconds are shows. As can be seen from Fig. 5, the smallest training time shown the common SGTM NLS.

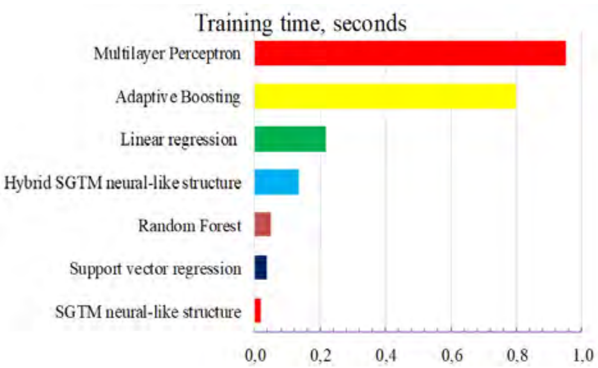


Fig. 5. Time comparison of different methods training procedures.

The proposed hybrid structure for providing the best result according to MAPE is 6 times slower than the classical SGTM neural-like structure (without polynomial inputs extension), but 7 times faster than the MPL.

The results of various methods in the form of the scatter plots [25] are visualized in Fig. 6. On the x-axis, the daily solar radiation is shown, and the results of the corresponding regression method are presented on the y-axis. In Fig. 6(*g*), it is also confirmed that the most accurate results of solar energy prediction are obtained by the developed method, based on the Hybrid SGTM NLS.

VII. Conclusion

In this paper, a developed approach to increase the approximation accuracy based on the use of computational intelligence tools is described. The Wiener polynomial usage for inputs extension of the neural-like structure greatly increases the accuracy of such tool. The non-iterative training procedure that is provided by the chosen computational intelligence tool maintains a high-performance training process. The compatible usage of these two approaches to solve a number of the task allows us to combine the above-mentioned advantages.

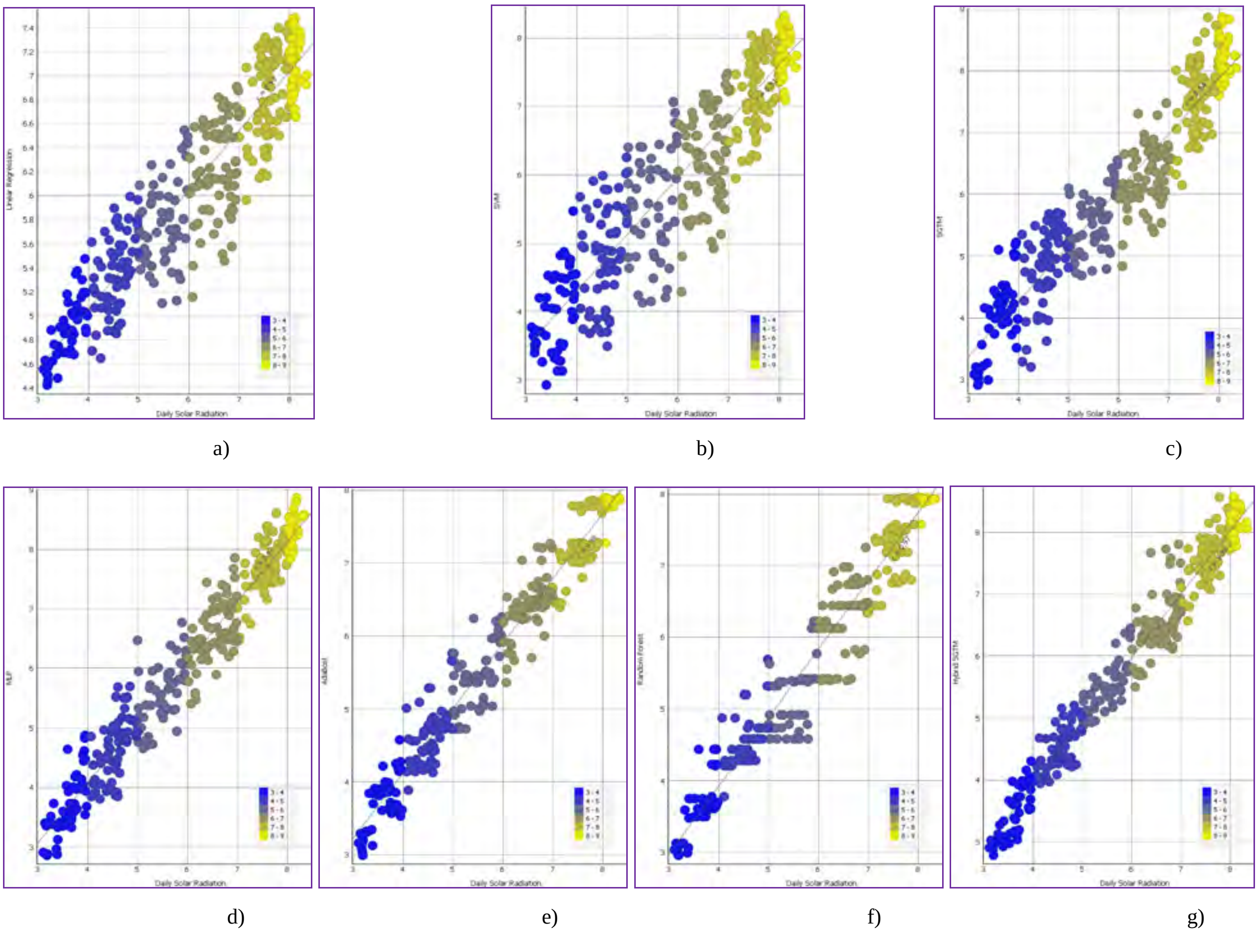


Fig. 6. Visual comparison of the various regression methods: a) Linear regression; b) Support Vector Regression; c) SGTM neural-like structure; d) Multilayer perceptron; e) AdaBoost; f) Random Forest; g) Hybrid SGTM neural-like structure.

The simulation of the proposed approach on real data shows the high accuracy of its work based on various indicators (MAPE, RMSE, R, Training time). Comparison of the results with modern methods (Support vector regression, Classic linear SGTM neural-like structure, Linear regression (with Stochastic Gradient Descent), Random Forest, Multilayer Perceptron, AdaBoost) shows its advantages, including the highest accuracy amongst all. At the same time, the time resources for implementing the training procedure in comparison with the classical methods are rather low.

The approach proposed in this paper can be used to solve a number of tasks that are critical to the time of performance (including training procedures) and performance indicators.

ACKNOWLEDGMENT

The authors' team are grateful to the reviewers for the relevant suggestions and comments that have significantly improved the quality of the article's presentation.

REFERENCES

[1] O. Mulesa, F. Geche, A. Batyuk, and V. Buchok, “Development of Combined Information Technology for Time Series Prediction,” Advances in Intelligent Systems and Computing II. CSIT 2017. Advances in Intelligent Systems and Computing, vol 689, Springer, Cham, pp. 361-373, 2018.

[2] Z. Hu, Y. Bodyanskiy, O. Tyshchenko and O. Boiko, “An Evolving Cascade System Based on a Set of Neo-Fuzzy Nodes,” International Journal of Intelligent Systems and Applications (IJISA), vol.8, no.9, pp.1-7, 2016.

[3] I. Dronyuk and O. Fedevych, “Traffic Flows Ateb-Prediction Method with Fluctuation Modeling Using Dirac Functions,” Computer Networks. CN 2017. Communications in Computer and Information Science, vol 718. Springer, Cham, pp. 3-13, 2017. doi.org/10.1007/978-3-319-59767-6_1

[4] V. Teslyuk, V. Beregovskyi, P. Denysyuk, T. Teslyuk and A. Lozynskyi, "Development and Implementation of the Technical Accident Prevention Subsystem for the Smart Home System," International Journal of Intelligent Systems and Applications(IJISA), vol.10, no.1, pp.1-8, 2018. DOI: 10.5815/ijisa.2018.01.01.

[5] N. Shakhovska and O. Shamuratov, "The structure of information systems for environmental monitoring," XIth Int. Scientific and Technical Conference Computer Sciences and Information Technologies (CSIT), Lviv, pp. 102-107, 2016. doi: 10.1109/STC-CSIT.2016.7589880

[6] I. Pliss and I. Perova “Diagnostic Neuro-Fuzzy System and Its Learning in Medical Data Mining Tasks in Conditions of Uncertainty about Numbers of Attributes and Diagnoses,” Automatic Control and Computer Sciences, vol. 51(6), pp. 391-398, 2017. DOI: 10.3103/S0146411617060062

[7] M. Nazarkevych, R. Oliiarnyk, H. Nazarkevych, O. Kramarenko and I. Onyshschenko “The method of encryption based on Ateb-functions,” In Data Stream Mining and Processing (DSMP), IEEE First International Conference, pp. 129-133, 2016.

[8] O. Riznik, I. Yurchak, E. Vdovenko and A. Korchagina, “Model of stegosystem images on the basis of pseudonoise codes,” VIth International Conference on Perspective Technologies and Methods in MEMS Design, Lviv, pp. 51-52, 2010.

[9] M. Nazarkevych, R. Oliarnyk, O. Troyan and H. Nazarkevych. “Data protection based on encryption using Ateb-functions,” In Scientific

and Technical Conference “Computer Sciences and Information Technologies (CSIT), pp. 30-32, 2016.

[10] J. Wainer, Comparison of 14 different families of classification algorithms on 115 binary datasets. arXiv:1606.00930. June 2016.

[11] Y. Bodyanskiy, O. Vynokurova, I. Pliss, G. Setlak and P. Mulesa, “Fast learning algorithm for deep evolving GMDH-SVM neural network in data stream mining tasks,” IEEE First International Conference on Data Stream Mining and Processing (DSMP), Lviv, pp. 257- 262. 2016. doi: 10.1109/DSMP.2016.7583555

[12] R. Tkachenko, , I. Izonin, “Model and Principles for the Implementation of Neural-Like Structures based on Geometric Data Transformations,” Advances in Computer Science for

Engineering and Education. ICCSEEA2018. Advances in Intelligent Systems and Computing. Springer, Cham, vol.754, pp.578-587, 2019. https://doi.org/10.1007/978-3-319-91008-6_58

[13] R. Tkachenko, P. Tkachenko, I. Izonin and Y. Tsymbal, “Learning-Based Image Scaling Using Neural-Like Structure of Geometric Transformation Paradigm”, Advances in Soft Computing and Machine Learning in Image Processing. Studies in Computational Intelligence, Springer, Cham, N. 1, vol. 730, pp. 537-567, 2018. doi.org/10.1007/978-3-319-63754-9_25

[14] U. Polishchuk, P. Tkachenko, R. Tkachenko and I. Yurchak, "Features of the auto-associative neurolike structures of the geometrical transformation machine," 5th Int. Conf. on Perspective Technologies and Methods in MEMS Design, Zakarpattya, Ukrane, pp. 66-67, 2009.

[15] I. Tsmots, V. Teslyuk, T. Teslyuk and I. Ihnatyev, “Basic Components of Neuronetworks with Parallel Vertical Group Data Real-Time Processing”, Advances in Intelligent Systems and Computing II. CSIT 2017. Advances in Intelligent Systems and Computing, vol. 689, Springer, Cham, pp. 558 – 576, 2018.

[16] O. Riznyk, I. Yurchak and O. Povshuk, “Synthesis of optimal recovery systems in distributed computing using ideal ring bundles”, XII Intern. Conf. on Perspective Technologies and Methods in MEMS Design (MEMSTECH), Lviv, pp. 220-222, 2016.

doi:10.1109/MEMSTECH.2016.7507545

[17] R. Tkachenko, Z. Duriagina, I. Lemishka, I. Izonin and A. Trostianchyn “Development of machine learning method of titanium alloys properties identification in additive technologies”, EasternEuropean Journal of Enterprise Technologies, Vol. 3, Iss. 12 (93), 2018, pp. 23-31. DOI: 10.15587/1729-4061.2018.134319

http://ijarcet.org/wp-content/uploads/IJARCET-VOL-5-ISSUE-4-919-923.pdf

[18] H. Kutucu and A. Almryad “Modeling of solar energy potential in Libya using an artificial neural network model” In: 2016 IEEE First International Conference on Data Stream Mining & Processing (DSMP), Lviv, 356-359, 2016.

[19] Z. Hu, Y. Bodyanskiy and O. Tyshchenko, “A Deep Cascade Neural Network Based on Extended Neo-Fuzzy Neurons and its Adaptive Learning Algorithm,” IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON), Kyiv, Ukraine, pp.801-805, May 29 – June 2, 2017.

[20] Y. Bodyanskiy, O. Vynokurova, I. Pliss and D. Peleshko, “Hybrid Adaptive Systems of Computational Intelligence and Their On-line Learning for Green IT in Energy Management Tasks,” Green IT Engineering: Concepts, Models, Complex Systems Architectures. Studies in Systems, Decision and Control, vol 74. Springer, Cham, 2017

[21] Y. Bodyanskiy, G. Setlak, D. Peleshko and O. Vynokurova, “Hybrid generalized additive neuro-fuzzy system and its adaptive learning algorithms,” IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Warsaw, pp. 328-333, 2015. doi: 10.1109/IDAACS.2015.7340753

[22] Ye. Bodyanskiy, O. Tyshchenko and D. Kopaliani, “Adaptive learning of an evolving cascade neo-fuzzy system in data stream mining tasks,” Evolving Systems, 7(2), pp.107-116, 2016.

[23] F. Geche, V. Kotsovsky, A. Batyuk, S. Geche and M. Vashkeba, "Synthesis of Time Series Forecasting Scheme Based on Forecasting Models System," 11th International Conference on ICT in Education, Research and Industrial Applications: Integration, Harmonization and Knowledge Transfer Lviv, Ukraine, May 14-16, CEUR Workshop Proceedings, vol. 1356, pp. 121-136, 2015

[24] I. Perova and Ye. Bodyanskiy, “Fast medical diagnostics using autoassociative neuro-fuzzy memory,” International Journal of Computing, 16 (1), 34-40, 2017.

[25] G. Setlak, Y. Bodyanskiy, O. Vynokurova and I. Pliss, "Deep evolving GMDH-SVM-neural network and its learning for Data Mining tasks," 2016 Federated Conference on Computer Science and Information Systems (FedCSIS), Gdansk, pp. 141-145, 2016.