UDC 004.052
DOI https://doi.org/10.32782/2663-5941/2023.5/39

Fomin O.O.
Odessa Polytechnic National University

Krykun V.A.
Odessa Polytechnic National University

Orlov A.A.
Odessa Polytechnic National University

Tataryn O.V.
Odessa Polytechnic National University

Litynskyi V.V.
Odessa Polytechnic National University

MATHEMATICAL MODELS OF SOFTWARE QUALITY ASSURANCE FOR INTERPRETATION OF DYNAMIC NEURAL NETWORKS

The article considers mathematical methods of software engineering in the problems of software quality assurance of intelligent systems. The indicators of functional suitability and freedom from risk in accordance with the international standard ISO/IEC 25010 are used as evaluation characteristics of software quality. The aim of the work is to improve the quality of dynamic neural network interpretation software using more adequate and accurate surrogate models in the form of functional series based on multidimensional weight functions. To achieve this goal, the following tasks were solved: the structures of neural networks for modeling nonlinear dynamic objects were investigated; analytical dependencies between the parameters of neural networks and multidimensional weight functions of the object were established; the method of constructing nonlinear dynamic models of time delay neural network interpretation in the form of functional series was further developed. The scientific novelty of the work lies in determining the information connection between time delay neural networks and functional series based on multidimensional weight functions. To reduce the computational volume of neural network training process, a linear rectification function is used as an activation function. To simplify the mathematical calculations, the linear rectification function is approximated by a polynomial in a certain interval. The practical utility of the work is to develop an algorithm for constructing surrogate models of nonlinear dynamic objects in the form of the functional series based on multidimensional weight functions by the results of training a time delay neural network. The practical significance of the obtained results is to improve the accuracy of neural network interpretation models used in intelligent systems software. The study of the proposed nonlinear surrogate model is carried out on the example of a test nonlinear dynamic object. The experiment demonstrates the advantages in accuracy of the surrogate models in the form of a functional series over the linear surrogate model.

Key words: *time-delay neural networks, surrogate models, nonlinear dynamic objects, software quality.*

Introduction. Computer system software has become extremely widespread in the modern world. Almost all branches of the national economy use computers with specialized software to solve a variety of tasks from office activities to applied tasks of control, management, diagnostics and prediction of the behavior of objects and processes of any nature. As the scope of specialized software use grows, so do the requirements for its quality. In these conditions, insufficient quality of software becomes not only a weakness but also a danger in critical areas of activity: production processes, transportation, medicine, etc. Therefore, ensuring the quality of software used in these areas is an important issue.

One of the areas of software engineering that has been developing rapidly in recent years is intelligent systems and machine learning. This area, of course, belongs to the critical areas of activity, so the issue of ensuring the quality of software for intelligent data processing is an urgent and, in general, unsolved problem.

The evaluation characteristics of software quality are regulated by the international standard ISO/IEC 25010. According to the standard, the set of characteristics and attributes of the quality model that reflect the requirements for the functioning of a software product is defined, among other things:

– Functional suitability: the ability of the software to solve the tasks required by users in certain conditions,

– Freedom from risk: the ability of the software to mitigate the negative effects of economic, health and safety risks.

Both characteristics depend on the adequacy and accuracy of the mathematical models used in machine learning.

The task of improving the quality of software for intelligent systems is complicated by the fact that the vast majority of modern control objects have nonlinear and dynamic properties, due to which they can function in more complex modes that cannot be realized using linear characteristics. In view of this, a significant improvement of these characteristics of the quality of software of intelligent systems can be achieved not so much with the help of technological and organizational means as through the development of mathematical and algorithmic methods of software engineering. This paper is devoted to the improvement of mathematical models of machine learning as an effective method of software quality assurance.

Statement of the problem. An analysis of recent research and publications has revealed a lack of mathematical methods for improving the quality of software of intelligent systems, including methods based on the construction of nonlinear surrogate models that interpret the work of neural network (NN) [9–11]. Therefore, the task of ensuring the quality of intelligent systems software by building interpretation models in the form of functional series based on multidimensional weight functions [4, 12] instead of linear surrogate models that replace NN is relevant and promising.

This work is aimed at eliminating the existing gap and is focused on the study of mathematical methods for improving the quality of intelligent systems software in the tasks of modeling dynamic objects with nonlinear characteristics and identifying the scope of their effective application in solving applied identification problems in critical areas of activity. This determines the purpose and objectives of this study.

Analysis of recent research and publications. In modern modeling of complex objects, NN are widely used. The convenience of using NN is due to the possibility of their construction only on the basis of measured data at the input and output of the object without any assumptions about the structure of the object and the internal laws of its functioning [1, 2]. Therefore, the use of NN to describe nonlinear dynamic objects, in particular, with continuous characteristics, is also becoming more widespread.

However, due to the high nonlinearity and complex interactions of a large number of parameters, NN do not explicitly reflect the structure and internal laws of the object's functioning [3]. As a result, NN do not provide transparency of the transformation of input data into output. Such structures are not convenient for studying the behavior of complex objects compared to analytical models, so finding ways to combine the advantages of NN and analytical models is a promising area of research to solve the problem of identifying complex objects [3, 4].

An effective way to improve the quality of software is to use more adequate and accurate mathematical models (surrogate models) that replace NN and, in some cases, interpret their behavior for humans.

Today, methods for building linear surrogate models are common [5, 6]. However, such models are of little use for identifying nonlinear objects. For a wide class of nonlinear dynamic objects, the relationship between the influence $x(t)$ and the response $y(t)$ can be explicitly represented by a functional (integral-step) series based on multidimensional weight functions [7, 8]. Due to the simultaneous consideration of nonlinear and inertial properties of the object, these models provide high adequacy of the control object and solution accuracy.

The aim of the article is to improve the quality of software for the interpretation of dynamic NN by using more adequate and accurate surrogate models in the form of functional series based on multidimensional weight functions.

To achieve this goal, the following tasks were set.

1. Selecting and studying the structure of NN for modeling nonlinear dynamic objects.

2. Establishing analytical dependencies between the parameters of the NN and the multidimensional weighting functions of the object.

3. Development of an algorithm for constructing a surrogate model in the form of a functional series based on multidimensional weight functions based on the results of NN training.

Dynamic models based on neural networks. Several methods are known for modeling nonlinear dynamic objects using NN [13]: dynamic neuro-spatial mapping (Dynamic Neuro-SM) [13–16], dynamic Wiener-type DNN [13, 17–19], and time-delayed NNs (TDNN) [12, 20, 21]. Among these model variants, TDNN is the most general structure consisting of several layers with direct signal propagation [4, 12, 21]. Such models are capable of learning from the input-output data of objects and have excellent convergence properties [4, 12, 21], which is an advantage over the aforementioned Dynamic Neuro-SM and Wiener-type DNN methods.

There are many TDNN structures: with multiple hidden layers, different activation functions, and topologies. In this paper, we consider a common TDNN structure consisting of three layers: input, hidden and output.

The TDNN input layer includes M neurons, where M is the memory size of the object model. The number of neurons M is chosen to best reflect the dynamic properties of the object [4, 12]. The layer receives input data $\mathbf{x}(t_n)=[x(t_n),\ x(t_{n-1}),\ \dots,\ x(t_{n-M-1})]$, $t_n=n\Delta t$, $n=1, 2, \dots$

The hidden layer includes K neurons with a nonlinear activation function. The number of neurons K is chosen to best reflect the nonlinear properties of the object.

The output layer includes 1 neuron with a linear activation function. The signal $y(t_n)$ at the output layer at time t_n depends both on the value of the input signal $x(t_n)$ at the current time t_n, and on the input data $x(t_{n-1}), \dots, x(t_{n-M-1})$ at times $t_{n-1}, \dots, t_{n-M-1}$. Thus, the output data $y(t_n)$ of the TDNN model is determined by the expression [12]:

$$y(t_n) = b_0 + S_0 \sum_{i=1}^{K} w_i S_i \left(b_i + \sum_{j=1}^{M} w_{i,j} x(t_{n-j}) \right) \qquad (1)$$

where b_0, b_i are the biases of the output and hidden layer neurons, respectively; S_0, S_i are the activation functions of the output and input layer neurons, respectively; w_i, $w_{i,j}$ are the weighting coefficients of the output and hidden layer neurons, respectively.

The most commonly used neuronal activation function S_i in the literature is the sigmoid and its derivatives, such as the hyperbolic tangent. Other activation functions can also be used, such as polynomial, sinusoidal, Gaussian, etc., or their combinations, depending on the purpose of a particular application.

The use of these functions has several advantages: they are nonlinear functions that approximate other nonlinear functions well; they are not discrete (stepwise), which makes activation analogous; they have smooth continuous derivatives.

Functional series based on multidimensional weight functions. Nonlinear systems with dynamic characteristics can be conveniently described using functional series based on multidimensional weight functions [4, 8, 12]. In the discrete form, functional series are used to describe an object with one input and one output in the time domain in the form of:

$$\begin{aligned} y(t_n) &= \sum_{p=0}^{\infty} \sum_{k=0}^{n} v_p(k,\dots,k) \prod_{i=1}^{p} x(t_n - k\Delta t) = \\ &= w_0 + \sum_{k=0}^{n} v_1(k) x(t_n - k\Delta t) + \sum_{k_1=0}^{n} \sum_{k_2=0}^{n} v_2(k_1,k_2) x(t_n - k_1\Delta t) x(t_n - k_2\Delta t) + \\ &+ \sum_{k_1=0}^{m} \sum_{k_2=0}^{m} \sum_{k_3=0}^{m} v_3(k_1,k_2,k_3) x(t_m - k_1\Delta t) x(t_m - k_2\Delta t) x(t_m - k_3\Delta t) + \dots \end{aligned} \qquad (2)$$

where $v_n(k_1,\dots,k_n)$ is a multidimensional weighting function of the p^{th} order in discrete form ($p=1,2,3,\dots$), symmetric with respect to the real variables $k_1,\dots,k_n$; n is the current count; $[0, n]$ is the summation interval, practically limited by the finite duration of the memory effect in the system.

The functional series based on multivariate weight functions (2) is a generalization of the power series to a functional space. The first term of the series is a well-known convolution integral, and the higher-order terms take into account higher-order dynamic nonlinearities. Thus, series (2) generalizes the convolution integral to the case of nonlinear objects.

The use of functional series makes it possible to take into account the nonlinear and inertial properties of an object more fully and accurately, makes the identification procedure more universal, and increases the accuracy of identification. These properties of models of nonlinear dynamic systems based on functional series have led to their widespread use in solving problems of modeling, identification, and synthesis of nonlinear systems.

Information connection of TDNN models and functional series. To date, there is no universal mathematical apparatus for converting NN into functional series. For the NN with the above-mentioned common activation functions (sigmoid, hyperbolic tangent, etc.), there have been attempts in the literature to build surrogate models in the form of functional series [12, 22].

But these functions have significant drawbacks. First, when moving away from the point $x=0$, *the* values of $S_i(x)$ of the sigmoidal function tend to react weakly to changes in the variable x. Consequently, the derivative in such regions takes small values, which leads to problems with the calculation of the gradient in the computer implementation of the learning algorithm: the gradient does not change due to extremely small values of the derivative. This leads to the fact that the NN refuses to learn further or does so extremely slowly [23, 24].

In order to reduce the computational burden and, therefore, speed up the training of NN, the linear rectification function **(**ReLU) is used as an activation function in practice:

$$S_i = \max(0, x) \qquad (3)$$

The ReLU function retains all the advantages of using a sigmoidal function: it is a nonlinear function that approximates other nonlinear functions well; it is not discrete (stepwise), which makes activation analog;

it has smooth continuous derivatives. In addition, this function is able to "dilute" neuronal activation. For example, for an NN with a large number of neurons, using a sigmoidal function as an activation function will cause all neurons to be activated to describe the network output. This is computationally expensive. If some neurons with negligible values of the activation function are excluded from the computational process – dilute the activation – the computational load will be significantly reduced, and the calculations will become more efficient. This is exactly what the activation function of ReLU can do**,** returning 0 for negative values of x [23, 24].

The disadvantage of a ReLU activation function NN is the difficulty in training. Most algorithms for training and optimizing NN parameters, including those based on the principle of backward error propagation, require a smooth activation function.

A well-known practice for solving this problem is to use a polynomial function that is as close as possible to the activation function in a certain interval [-q, q]:

$$S_i = \sum_{p=0}^{H} a_p x^p \tag{4}$$

where p is the order of the polynomial, $p=0$,1,2,...

If we approximate function (3) with a polynomial (4) on a certain interval [-q, q] and the function $S_0 = x$ to simplify mathematical calculations, the expression for the network model (1) can be written as follows:

$$y(n) = b_0 + \sum_{i=0}^{K} w_i \sum_{p=0}^{H} a_p \left(b_i + \sum_{j=0}^{M} w_{i,j} x(n-j) \right)^p \tag{5}$$

Analyzing expressions (2) and (5), we can conclude that they are isomorphic with respect to each other. Therefore, to build a surrogate model for TDNN, it is necessary to establish an information link between the functional series (2) and the TDNN models (5).

Thus, by setting the degree of the approximating polynomial H, it is possible to obtain the information connection of models in the form of TDNN and functional series in an analytical form.

p=0*:*

$$y(n) = b_0 + a_0 \sum_{i=0}^{K} w_i \tag{6}$$

$$v_0 = b_0 + a_0 \sum_{i=0}^{K} w_i \tag{7}$$

p=1*:*

$$y(n) = b_0 + \sum_{i=0}^{K} w_i (a_0 + a_1 b_i) + a_1 \sum_{i=0}^{K} w_i \sum_{j=0}^{M} w_{i,j} x(n-j) \tag{8}$$

$$v_0 = b_0 + \sum_{i=0}^{K} w_i (a_0 + a_1 b_i); \quad v_1 = a_1 \sum_{i=0}^{K} w_i \sum_{j=0}^{M} w_{i,j} \tag{9}$$

p=2*:*

$$y(n) = b_0 + \sum_{i=0}^{K} w_i (a_0 + a_1 b_i + a_2 b_i^2) + \sum_{i=0}^{K} w_i (a_1 + 2a_2 b_i) \sum_{j=0}^{M} w_{i,j} x(n-j) + a_2 \sum_{i=0}^{K} w_i \sum_{j=0}^{M} w_{i,j}^2 x^2(n-j) \tag{10}$$

$$v_0 = b_0 + \sum_{i=0}^{K} w_i (a_0 + a_1 b_i + a_2 b_i^2); \quad v_1 = \sum_{i=0}^{K} w_i (a_1 + 2a_2 b_i) \sum_{j=0}^{M} w_{i,j}; \quad v_2 = a_2 \sum_{i=0}^{K} w_i \sum_{j=0}^{M} w_{i,j}^2 \tag{11}$$

p=3*:*

$$y(n) = b_0 + \sum_{i=0}^{K} w_i (a_0 + a_1 b_i + a_2 b_i^2 + a_3 b_i^3) + \sum_{i=0}^{K} w_i (a_1 + 2a_2 b_i + 3a_3 b_i^2) \sum_{j=0}^{M} w_{i,j} x(n-j) + \sum_{i=0}^{K} w_i (1 + 3a_3 b_i) \sum_{j=0}^{M} w_{i,j}^2 x^2(n-j) + a_3 \sum_{i=0}^{K} w_i \sum_{j=0}^{M} w_{i,j}^3 x^3(n-j) \tag{12}$$

$$v_0 = b_0 + \sum_{i=0}^{K} w_i (a_0 + a_1 b_i + a_2 b_i^2 + a_3 b_i^3); \quad v_1 = \sum_{i=0}^{K} w_i (a_1 + 2a_2 b_i + 3a_3 b_i^2) \sum_{j=0}^{M} w_{i,j}; \quad v_2 = \sum_{i=0}^{K} w_i (1 + 3a_3 b_i) \sum_{j=0}^{M} w_{i,j}^2; \quad v_3 = a_3 \sum_{i=0}^{K} w_i \sum_{j=0}^{M} w_{i,j}^3 \tag{13}$$

Expressions (7), (9), (11), (13) are estimates of multivariate weight functions of orders $p=0$, 1, 2, 3, respectively, obtained using TDNN**.** Similarly, expressions for estimates of multivariate weight functions of higher orders can be obtained, but in practice, models in the form of functional series of higher orders are rarely used.

Thus, the estimates of multidimensional weight functions can be expressed in terms of the neuronal bias values b_0, b_j and weight coefficients w_i, $w_{i,j}$ ($i=1,...,K$, $j=1,...,M$) of the output and hidden layers of the NN, respectively, and the coefficients a_p of the polynomial approximating the activation function in the hidden layer.

The accuracy of the approximation of the polynomial function (4) is closely related to the quality of the NN. Therefore, to calculate the polynomial approximation of function (3) on the set {x_i , max(0, x_i)} (where x_i are random values on the interval [-q, q] with a normal distribution), a least-squares regression method is used [25].

To calculate the polynomial approximation of the ReLU on the standard normal distribution, we use the polynomial function *Polynomial.fit* from the Python numpy package. Using this function for the input data {x_i , max(0, x_i)} and different orders p of the approximation function gives sets of coefficients a_i , i=1,...,p. The results of approximating the ReLU function using polynomials of different orders with a standard normal distribution of 99.73% on the interval [-3, 3] are shown in Table 1.

A method for constructing a nonlinear dynamic surrogate model. The significant practical value of the constructed connection of TDNN-based models and functional series based on multidimensional weight functions lies in the further development of the method for estimating multidimensional weight functions directly from the parameters of the NN [4, 12, 22]. This method is useful in the tasks of ensuring the quality of intelligent systems software by using more adequate and accurate surrogate models in the form of functional series based on multidimensional weight functions.

The algorithm of the method of constructing surrogate models in the form of functional series based

Table 1

Approximation of the linear rectification function using polynomials of different orders

The degree of a polynomial	Polynomial S_i	Accuracy approximations
1	$0.85714286 + 1.5\,x$	1.857
2	$0.1992 + 0.5002\,x + 0.1997\,x^2$	0,279
3	$0.1995 + 0.5002\,x + 0.1994\,x^2 – 0.0164\,x^3$	0,161
4	$0.1298 + 1.500\text{-}x + 2.5909\text{-}x^2 – 0.0001\,x^3 – 1.2272\text{-}x^4$	0.038

on the results of training TDNN with the activation function ReLU takes the following sequence of steps:

Step 1. Determine the model memory size M, the number of neurons in the hidden layer K, and the modeling accuracy e.

Step 2. Set the set $\mathbf{D}(t)=\{\mathbf{x}(t_n), y(t_n)\}$ – training set, $n=1,...,N$, where N is the number of measurements in the input-output experiment; define the operation of preliminary data normalization by the expression:

$$\mathbf{D}'(t)=[(\mathbf{D}(t)\text{-}mean(\mathbf{D}(t)))/(max(\mathbf{D}(t))\text{-}min(\mathbf{D}(t))]. \quad (14)$$

Step 3. Take the number of training iteration $s=1$, initialize the shift vectors $\mathbf{B}=[b_0, b_j]$ and the weighting coefficients $\mathbf{W}=[w_i, w_{i,j}]$ $(i=1,...,K,\ j=1,...,M)$ using a random value on the interval (0,1).

Step 4. Determine, based on one of the back propagation algorithms, the bias errors b_0, b_j and the weighting factors w_i, $w_{i,j}$ of the NN.

Step 5. Check the conditions for completing the training. If $e_s(t) \le e$ go to *Step 6*, otherwise $s=s+1$ and go to *Step 4*.

Step 6. Determine the multidimensional weighting functions using expressions (6)-(13).

The above algorithm can be compactly written using pseudocode:

Algorithm: *surrogate_nn_model*
Input: M, K, ε, E, N, $\mathbf{D}$
Output: $\mathbf{V}_n$
normalize ← **D**
$\mathbf{D}_{train}$, $\mathbf{D}_{test}$ ← **D**
random ← **B**[*K*], **W**[*M*, *K*]
epoch ← 0
while *epoch < E or* $e_s(t) > e$ **do**
epoch ← *epoch + 1*
for *i = 1, ..., N* **do**
evaluations ← *training procedure(***B**, **W**, $\mathbf{D}_{train}$, *e)*
end for
loss ← *mse*(**B**, **W, D** $)_{test}$
end while
V← *bwf*(**B**, **W**)

Investigation of the accuracy of a nonlinear surrogate model in the form of a functional series. The study of the proposed nonlinear surrogate model is carried out on the example of a test nonlinear dynamic object. The simulation model of the test object with a first-order dynamic block and a nonlinear feedback block [4] is shown in Fig. 1.

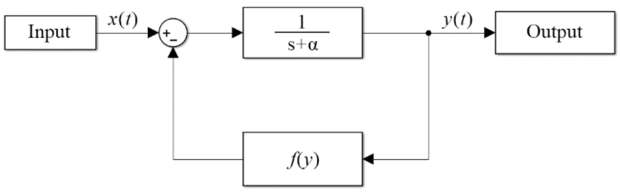

Fig. 1. Simulation model of the test nonlinear dynamic object

The feedback block uses a nonlinear function with saturation as $f(y)$:

$$f_2(y) = \begin{cases} s, & y > p \\ k \cdot y, & |y| \le p \\ -s, & y < -p \end{cases} \quad (15)$$

where s is the saturation level, p is the saturation start point, and $k=s/p$ is the gain. The following parameters of the simulation model were adopted: α=2.64; s=0.7, p=0.7, k=1.

To determine the accuracy of the proposed surrogate model in the form of a functional series based on multidimensional weight functions $y_v(t)$, the test nonlinear dynamic object is identified based on the results of input/output experiments. The obtained results are compared with the simulation model $y(t)$, the neural network model $y_l(t)$, and the linear surrogate model $y_l(t)$, built on the same data (Fig. 2).

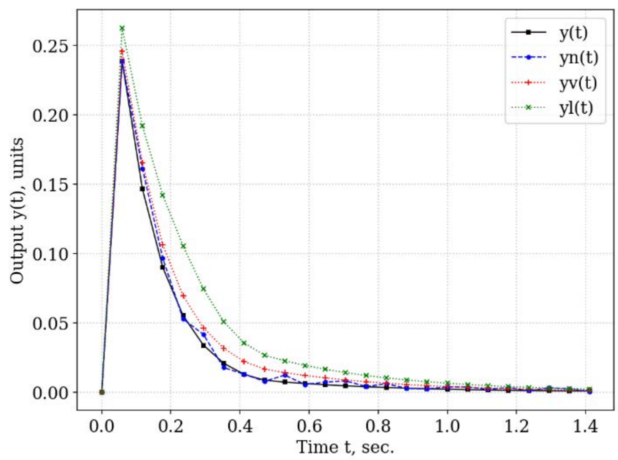

Fig. 2. Comparison of the surrogate model in the form of a functional series $y_v(t)$ with the simulation model $y(t)$, the neural network model $y_l(t)$, and the linear surrogate model $y_l(t)$

The experiment demonstrates the advantages in accuracy of the surrogate model in the form of a functional series $y_v(t)$ over the linear surrogate model $y_l(t)$.

Conclusions. The work is devoted to the study of mathematical methods for improving the quality of intelligent systems software in the tasks of modeling dynamic objects with nonlinear characteristics and identifying the scope of their effective application in solving applied identification problems in critical areas of activity.

To model a dynamic object with nonlinear characteristics, the structure of an artificial neural network with time delays is substantiated. The analytical dependence between the weight coefficients of the neural network with time delays and the multidimensional weight functions of a nonlinear object is established.

The method of constructing nonlinear dynamic models for the interpretation of neural networks with time delays in the form of a functional series has been further developed. The advantage of the developed method in comparison with the existing ones is the increase in the training speed of a neural network with time delays by using the activation function of linear rectification.

The algorithm of the method of constructing a surrogate model in the form of a functional series based on multidimensional weight functions in the form of pseudo-code of the software of an intellectual system is created.

The created algorithm makes it possible to build nonlinear surrogate models that have advantages in accuracy over linear surrogate models.

Bibliography:

1. N. Todorovic and P. Klan, "State of the Art in Nonlinear Dynamical System Identification using Artificial Neural Networks," 2006 8th Seminar on Neural Network Applications in Electrical Engineering, Belgrade, Serbia, 2006, pp. 103-108.
2. C. Rudin and J. Radin, “Why are we using black box models in AI when we don’t need to? A lesson from an explainable AI competition”, Harvard Data Science Review, vol. 2, no. 1, 2019.
3. A. R. Rao and M. Reimherr, “Non-linear functional modelling using neural networks”, 2021. (https://arxiv.org/abs/2104.09371)
4. Fomin, O., Polozhaenko, S., Krykun, V., Orlov, A., Lys, D. (2023). Interpretation of Dynamic Models Based on Neural Networks in the Form of Integral-Power Series. In: Arsenyeva, O., Romanova, T., Sukhonos, M., Tsegelnyk, Y. (eds) Smart Technologies in Urban Engineering. STUE 2022. Lecture Notes in Networks and Systems, vol 536. Springer, Cham, pp. 258-265.
5. Md. Rezaul Karim, Md. Shajalal, A. Graß, T. Döhmen, S.A. Chala, Ch. Beecks, S. Decker. Interpreting Black-box Machine Learning Models for High Dimensional Datasets. 2022. (https://arxiv.org/abs/2208.13405)
6. A. Agresti, “Foundations of linear and generalized linear models”, Wiley series in probability and statistics, 2017.
7. J. Schoukens and L. Ljung, “Nonlinear System Identification: A User-Oriented Road Map”, IEEE Control Systems Magazine, vol. 39, no. 6, 2019, pp. 28-99.
8. Pavlenko, V.D. and Pavlenko, S.V. 2023. Deterministic identification methods for nonlinear dynamical systems based on the Volterra Model. Applied Aspects of Information Technology. 1, 1 (Jun. 2023), pp. 11-32.
9. Murdoch, W. J., Singh, C., Kumbier, K., et al. (2019). Defnitions, methods, and applications in interpretable machine learning. Proceedings of the National Academy of Sciences, 116(44), 22071– 22080. https://doi.org/10.1073/pnas.1900654116.
10. R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, D. Pedreschi, A survey of methods for explaining black box models, in: ACM Computer Surveys, volume 51, no. 5, 2018, pp. 1–42. doi: 10.1145/3236009.
11. C. A. Mitrea, C. K. M. Lee, Z. Wu, A comparison between neural networks and traditional forecasting methods: case study, in: International journal of engineering business management, volume 1, no. 2, 2009, pp. 19–24. doi: 10.5772/6777.
12. G. Stegmayer, M. Pirola, G. Orengo, O. Chiotti, Towards a Volterra series representation from a neural network model, in: WSEAS Transactions on Circuits and Systems, archive 1, 2004, pp. 55–61.
13. W. Liu, W. Na, L. Zhu and Q. -J. Zhang, "A review of neural network based techniques for nonlinear microwave device modeling," 2016 IEEE MTT-S International Conference on Numerical Electromagnetic and Multiphysics Modeling and Optimization (NEMO), Beijing, China, 2016, pp. 1-2.
14. Wenyuan Liu, L. Zhu, Weicong Na and Q. -J. Zhang, "An overview of Neuro-space mapping techniques for microwave device modeling," 2016 IEEE MTT-S Latin America Microwave Conference (LAMC), Puerto Vallarta, Mexico, 2016, pp. 1-3.
15. L. Zhu, Q. Zhang, K. Liu, Y. Ma, B. Peng and S. Yan, "A Novel Dynamic Neuro-Space Mapping Approach for Nonlinear Microwave Device Modeling," in IEEE Microwave and Wireless Components Letters, vol. 26, no. 2, 2016, pp. 131-133.

16. W. Liu, Y. Su and L. Zhu, "Nonlinear Device Modeling Based on Dynamic Neural Networks: A Review of Methods," 2021 IEEE 4th International Conference on Electronic Information and Communication Technology (ICEICT), Xi'an, China, 2021, pp. 662-665.

17. W. Liu, Y. Su, H. Tan, F. Feng and B. Zhang, "A Review of Wiener-Type Dynamic Neural Network for Nonlinear Device Modeling" 2022 IEEE MTT-S International Microwave Workshop Series on Advanced Materials and Processes for RF and THz Applications (IMWS-AMP), Guangzhou, China, 2022, pp. 1-3.

18. W. Liu, W. Na, L. Zhu, J. Ma and Q.-J. Zhang, "A Wiener-Type Dynamic Neural Network Approach to the Modeling of Nonlinear Microwave Devices," in IEEE Transactions on Microwave Theory and Techniques, vol. 65, no. 6, 2017, pp. 2043-2062.

19. W. Liu, W. Na, F. Feng, L. Zhu and Q. Lin, "A Wiener-Type Dynamic Neural Network Approach to the Modeling of Nonlinear Microwave Devices and Its Applications," 2020 IEEE MTT-S International Conference on Numerical Electromagnetic and Multiphysics Modeling and Optimization (NEMO), Hangzhou, China, 2020, pp. 1-3.

20. M. Sugiyama, H. Sawai and A. H. Waibel, "Review of TDNN (time delay neural network) architectures for speech recognition", IEEE International Sympoisum on Circuits and Systems, vol. 1, 1991, pp. 582–585.

21. L. Wenyuan, L. Zhu, F. Feng, W. Zhang, Q.-J. Zhang, L. Qian and G. Liu, "A time delay neural network based technique for nonlinear microwave device modelling, in: Micromachines", Basel, vol. 11, no. 9, 2020, p. 831.

22. Ruban, O. Volterra neural network construction in the nonlinear dynamic systems modeling problem / O. Ruban // Herald of Advanced Information Technology. – Odesa, 2019. – Vol. 2, N 1. – P. 24–28.

23. J. Pomerat, A. Segev and R. Datta, "On Neural Network Activation Functions and Optimizers in Relation to Polynomial Regression," 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 2019, pp. 6183-6185, doi: 10.1109/BigData47090.2019.9005674.

24. Ramy E. Ali, Jinhyun So and A. Salman Avestimehr. On Polynomial Approximations for Privacy-Preserving and Verifiable ReLU Networks. 2021. (https://arxiv.org/abs/2011.05530)

25. Chabanne, H., Wargny, A.D., Milgram, J., Morel, C., & Prouff, E. Privacy-Preserving Classification on Deep Neural Network.

Фомін О.О., Крикун В.А., Орлов А.А., Татарин О.В., Літинський В.В. МАТЕМАТИЧНІ МОДЕЛІ ЗАБЕЗПЕЧЕННЯ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІНТЕРПРЕТАЦІЇ ДИНАМІЧНИХ НЕЙРОННИХ МЕРЕЖ

Розглядаються математичні методи інженерії програмного забезпечення в задачах забезпечення якості програмного забезпечення інтелектуальних систем. В якості оціночних характеристик якості програмного забезпечення прийняті показники функціональної придатності та свободи від ризику згідно міжнародним стандартом ISO/IEC 25010. Метою роботи є підвищення якості програмного забезпечення інтерпретації динамічних нейронних мереж шляхом використання більш адекватних та точних сурогатних моделей у вигляді функціональних рядів на основі багатовимірних вагових функцій. Для досягнення поставленої мети вирішено наступні завдання: досліджено структури нейронних мереж для моделювання нелінійних динамічних об'єктів; встановлено аналітичні залежності між параметрами нейронними мережами та багатовимірними ваговими функціями об'єкту; набув подальшого розвитку метод побудови нелінійних динамічних моделей інтерпретації нейронних мереж з часовими затримками у вигляді функціональних рядів. Наукова новизна роботи полягає у визначенні інформаційного зв'язку між нейронними мережами з часовими затримками та функціональними рядами на основі багатовимірних вагових функцій. Для зниження обчислювального навантаження навчання нейронну мережу в якості активаційної функції використовується функція лінійної ректифікації. Для спрощення математичних викладок функцію лінійної ректифікації апроксимовано поліномом на певному інтервалі. Практична користь роботи полягає у розробці алгоритму методу побудови сурогатних моделей нелінійних динамічних об'єктів у вигляді функціонального ряду на основі багатовимірних вагових функцій за результатами навчання нейронної мережі з часовими затримками. Практичне значення одержаних результатів полягає у підвищенні точності моделей інтерпретації нейронних мереж, що використовуються в програмному забезпеченні інтелектуальних систем. Дослідження за-пропонованої нелінійної сурогатної моделі проведено на прикладі тестового нелінійного динамічного об'єкта. Експеримент демонструє переваги в точності сурогатної моделі у вигляді функціонального ряду над лінійною сурогатною моделлю.

Ключові слова: *нейронні мережі з часовими затримками, сурогатні моделі, нелінійні динамічні об'єкти, якість програмного забезпечення.*