Method of Support Neural Networks for Modeling Nonlinear Dynamics

Oleksandr Fomin[1,*,†], Sergii Polozhaenko[2,†], Andrii Prokofiev[3,†] and Oleksiy Tataryn[4,†]

[1,2,3,4] *Odesa Polytechnic National University, 1, Shevchenko ave., Odesa, 65044, Ukraine*

Abstract

The work is devoted to resolving the contradiction between the speed of modeling nonlinear dynamic objects and the accuracy of constructing models in the form of neural networks. The purpose of the work is to reduce the time required to build nonlinear dynamics continuous-time models in the form of neural networks while ensuring the specified modeling accuracy. This purpose is achieved by developing a modeling method based on the superposition of a set of pre-trained neural networks (support models) that reflect the main properties of the subject area. The scientific novelty of the developed method lies in using pre-trained neural networks with time delays as support models for modeling nonlinear dynamic objects. Unlike existing pre-training methods, the developed method allows building simpler models with reduced training time while ensuring the specified accuracy. The practical benefit of the results of the work lies in the development of the support models method algorithm, which allows significantly reducing the training time of neural networks with time delays without losing the accuracy of the model.

Keywords

identification; nonlinear dynamics; pre-training; neural networks training speed

1. Introduction

The current stage of modeling development, which is mainly based on using intelligent technologies, is marked by a number of requirements from practice both for high accuracy of models and for the speed of their construction [1].

Achieving high accuracy of nonlinear dynamics continuous-time modeling today is carried out through using machine learning methods, in particular, neural networks (NN) [2-4]. However, applying such methods is often associated with high computational complexity, which leads to significant time spent on building models [3-5].

The problem of increasing the speed of modeling remains one of the most urgent, especially in industries related to the personalization of models, which must adapt to changes in user behavior or the environment (e.g., in authentication tasks, biomedical applications, human-machine systems), while operating in real time [6, 7].

It should be noted that various approaches are being actively researched as part of efforts to accelerate NN learning. The most common of these are based on regularization and normalization methods, which promote faster convergence and reduce the likelihood of overfitting [8]. Popular experiments include activation functions aimed at reducing computational complexity and [9] model construction time [10] and experiments on optimizing NN architecture (LSTM, RC networks, etc.), which demonstrate a significant reduction in training time while maintaining high performance for dynamic systems. These studies emphasize the ongoing search for methods to improve the efficiency and speed of building intelligent models. One of the common approaches to accelerating the process

ICST-2025: Information Control Systems & Technologies, September 24-26, 2025, Odesa, Ukraine

* Corresponding author.

† These authors contributed equally.

fomin@op.edu.ua (O. Fomin); polozhaenko@op.edu.ua (S. Polozhaenko); fallbrick1985@gmail.com (A. Prokofiev); otataryn@stud.op.edu.ua (O. Tataryn)

0000-0002-8816-0652 (O. Fomin); 0000-0002-4082-8270 (S. Polozhaenko); 0000-0002-4520-4248 (A. Prokofiev); 0009-0005-3888-6569 (O. Tataryn)

of building an NN is pre-training [11]. Pre-trained models can be quickly adapted to new tasks, making them an effective tool in reducing simulation time.

This direction seems promising in modeling objects with a high degree of internal complexity and interaction, first of all, nonlinear dynamic objects [2]. At the same time, there is a significant lack of studies in the field of NN pre-training that simulate the nonlinear dynamics of continuous objects.

The purpose of the work is to reduce the time for building nonlinear dynamics continuous-time models in the form of NN while ensuring the specified modeling accuracy by developing a method based on the NN models pre-training.

To reach this goal, the following tasks were formulated.

1. Development of the modeling method based on pre-training by superposition of a set of pre-trained NN (support models) reflecting the basic characteristics of the subject area.

2. Building of support models in the form of NN reflecting the basic nonlinear and dynamic characteristics of the subject area.

3. Study of the speed of modeling complex nonlinear dynamics using the developed method of support models.

2. Problem statement

The approach to pre-training NN in practice faces a number of significant limitations [12, 13]. A characteristic feature of the pre-training process is a significant time spent on building a general model [14-16]. To achieve the purpose of the study, it is necessary to invent a way to accelerate the construction of general models of nonlinear dynamics. The formal formulation of the problem of accelerating the construction of general models of nonlinear dynamics based on pre-training of NN is as follows.

Let S be the domain for which there is enough labeling data N_S (D_S dataset):

$$D_S=\{(x_i^S, y_i^S)\}, \quad (1)$$

where x_i^S is a vector of independent variables, y_i^S is the corresponding target variable (label), $i=1,\ldots, N_S$.

Let $f_{\theta S}$ be the general NN model with θ_S parameters, which is trained on D_S dataset.

Let T be the target task in the subject area S, for which there are marked-up data of limited N_T size (D_T dataset):

$$D_T=\{(x_j^T, y_j^T)\}, \quad (2)$$

where x_j^T is the vector of the independent variables, y_j^T is the corresponding target variable (label), $j=1,\ldots, N_T$.

Let $f_{\theta T}$ be the target NN model with θ_T parameters, trained on D_T dataset, which ensures the accuracy $E_{\theta T}$ and the duration $t_{\theta T}$ of target model training.

It is necessary to find the parameters θ_S of the general model. Using them as starting values θ_{T0} during training, the target model $f_{\theta T}$, a specified accuracy level $E_{\theta T}$ is achieved in the minimum time $t_{\theta T}$:

$$\theta_{T0}=\theta_S : \underset{t_{\theta T}}{arg\ min} L_T(f_{\theta T}(\mathbf{x}_j^T), y_j^T) = E_{\theta T} \quad (3)$$

where L_T is the loss function adopted for the target model.

3. Method of support models

3.1. The method of support models

To accelerate the construction of general models of nonlinear dynamics, a new approach is proposed in the work. It consists in using a set of separate pre-trained NN (support models), each of which reflects separate basic characteristics of the domain.

To construct a set of support models, a set of support datasets D_{Rk} is used (k=1,...,g, g is the number of basic characteristics of the domain). Each of the support datasets D_{Rk} describes a separate basic characteristic of the domain. On the basis of these datasets, support models $f_{\theta Rk}$ with the parameters θ_{Rk} are built. By combining support models that reflect the characteristics of the target problem, a general model $f_{\theta S}$ is built. It has a set of specified properties (nonlinear and dynamic). The target model $f_{\theta T}(\theta_S, D_T)$ is built by pre-training the general model $f_{\theta S}$ obtained on the basis of the set of support models $f_{\theta Rk}$ on the D_T dataset.

This approach allows to preserve the advantages of pre-training, since support models obtained once can be repeatedly used for different domains and target tasks, significantly reducing the total time and resources for training models without collecting additional data [17-19].

The structural scheme of the training process based on support models is presented in Fig. 1.

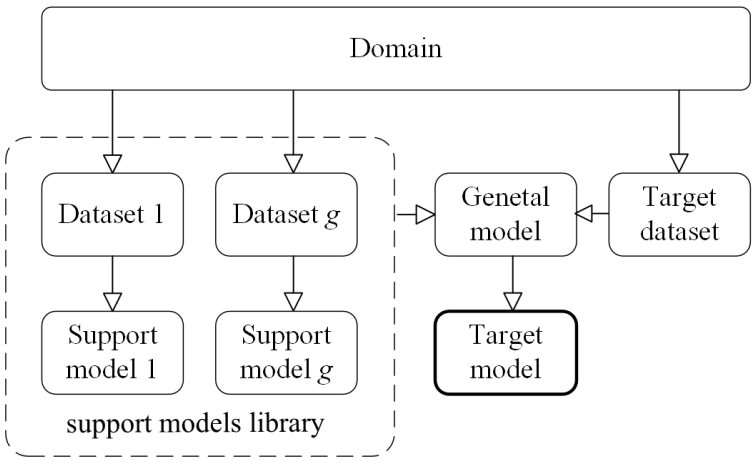

Figure 1: Structural diagram of the training process using support models

The algorithm of the suggested method consists of the following steps.

Step 1. Selection of basic domain properties and formation of a set of datasets D_{Rk} reflected the selected properties.

Step 2. Construction of support models set $f_{\theta Rk}$ in the form of separate NN corresponding to the established properties of the domain. Training of built models based on the generated datasets D_{Rk}.

Step 3. Determination of the list of p properties of the target problem from the set of g basic properties of the domain and construction of the general model $f_{\theta S}$ based on the superposition of the corresponding support models $f_{\theta Rh}$ (h=1,...,p, $p \le g$) obtained in *Step* 2.

Step 4. Training of the target model $f_{\theta T}(\theta_S, D_T)$ based on the general model $f_{\theta S}$ obtained in *Step* 3.

Step 5. Determination of the accuracy indicators $E_{\theta T}$ and training time $t_{\theta T}$ of the target model $f_{\theta T}$. In case of unsatisfactory quality indicators of the target model $f_{\theta T}$, the control transfers to *Step* 2 to correct the structure θ_{Rk} of the support models $f_{\theta Rk}$, and, if necessary, to *Step 1* to correct the set of basic properties of the domain and the set of datasets D_{Rk}, reflecting the selected properties.

3.2. Selecting the basic properties of the domain for forming datasets

The basic properties of the domain in the work are understood as the characteristics of objects that reflect the essential features of their behavior. These properties can include real or abstract parameters that are important for solving the modeling problem [20].

The procedure for selecting the basic properties of the domain and generating datasets D_{Rk} reflecting the selected properties is as follows.

1. Defining the range of tasks to be solved in a given domain; analyzing properties that are important for objects in a given domain, significantly affect the results of modeling and should be taken into account when forming the dataset D_S.

2. Determination of signal types (for example, periodic, random, pulsed) that best reflect the properties of the objects under study.

3. Formation of the dataset D_S based on the list of basic properties of the domain established in paragraph 1, the set of input signals and reactions of the object formed in paragraph 2.

4. Segmentation of the dataset D_S into separate datasets D_{Rk} according to the defined list of basic properties of the domain.

In the above sequence of steps, the task of determining the type of signals that best reflect the properties of the object under study remains the least formalized. Automating the selection of support models is crucial for scaling the method to complex, multidimensional dynamical systems and bridges the gap between human intuition and automated processing, allowing the application of machine learning methods for objective identification and segmentation.

To automate the selection of support models, improve the reproducibility and reduce the subjective factor of the support model method, this paper proposes two approaches: input data clustering [21] and feature contribution analysis [22].

1. Clustering-based approach: to automatically segment the overall D_S dataset into D_{Rk} subsets that clearly demonstrate elementary properties of the object under study.

2. Feature contribution analysis approach: to identify and quantify the underlying properties of the subject area P_k.

3.3. Determination of the structure of support models and their pre-training

To modelling nonlinear dynamics, the work uses time-delayed NN (TDNN) [23]. Due to their simplicity and versatility, TDNNs are most widely used in modeling problems of nonlinear dynamic objects. In practice, the TDNN structure is most often used, consisting of three layers: input, hidden, and output [24]. The size of the layers in this TDNN structure is determined as follows:

- the input layer consists of M neurons and is responsible for the memory (dynamic characteristics) of the model,
- the hidden layer consists of K neurons and is responsible for the nonlinear characteristics of the model,
- the output layer contains the number of Y neurons, which is equal to the number of outputs of the model.

For each support model, a labeled data set $D_{Si}=\{(x^{Pk}_{Hj}(t), f_{Vi}[x^{Pk}_{Hj}(t)])\}$ is formed based on the signals $x(t)$ at the input of the object and the responses $y(t)= f_{Vi}[x(t)]$ at its output. Typical signals are often used as input signals: impulse $x(t)=a\delta(t)$, step $x(t)=a\Theta(t)$, linear $x(t)=at$, and harmonic $x(t)=a\ sin(t)$ signals of various amplitudes $a \in (0, 1]$. Time delays at the input are implemented through shifts in time series and the inclusion of previous values in the input vector.

The main steps for implementing time delays and forming a training sample are as follows:

- Step 1. Selecting the number of delays. Determine the number of delays M that will be used.
- Step 2. Forming the input vector. For each time moment tk, the input vector is formed as a sequence of current and previous values (2.1).
- Step 3. Transferring delays to the network. Each formed input vector is transmitted to the input layer of the neural network, where it is processed as a regular input.

The algorithm for forming a training sample with time delays in pseudocode is shown below.

Algorithm 1: time_delay

1: *Input*: D_{Si}, M, $x(t)$, $y(t)$
2: *Output*: x, y
3: *foreach* D_{Si} as $x(t)$, $y(t)$
4: *for* k =0, ..., T-M+1 *do*
5: $x_k \leftarrow [x(t_k), x(t_{k-1}), x(t_{k-2}), ..., x(t_{k-M+1})]$
6: $y_k \leftarrow y(t_k)$
7: *end for*
9: *end foreach*

3.4. Construction of a general model based on the superposition of the corresponding support models

After completing the process of pre-training the set of support models $f_{\theta Rk}$, a general model $f_{\theta R}$ is built on their basis. This model is composed of a set of p support models $f_{\theta Rh}$ that correspond to the available basic properties of the object. The selection of support models that reflect the basic properties of an object is generally subjective. To reduce subjectivity and increase the reproducibility of the selection of support models, it is advisable to use methods of clustering input data or feature contribution analysis.

After completing the preliminary training process for the family of support models that correspond to the existing basic characteristics of the object, a rough model $f_{\theta S}(D_S)$ is constructed based on them. Assuming that the general $f_{\theta S}$ and support $f_{\theta Rk}$ models are constructed in the form of NN with the same structure (dimension of the parameter vector $dim(\theta_S)=dim(\theta_{Rk})$), the definition of the general model is reduced to calculating the arithmetic operations on corresponding components of the parameter vectors of the support models θ_{Rk}.

At the same time, several approaches to the superposition of support models are considered:

- additive superposition is used when each support model is responsible for independent aspects of the system (e.g., dynamics and environmental impact). The outputs of the support models fθSi(DSi) are determined based on the calculation of the arithmetic mean of the corresponding components of the parameter vectors of the support models θSv:

$$\theta_S^i = \frac{1}{h}\sum_{k=1}^{h}\theta_{Rk}^i \tag{4}$$

  where i are the indices of the corresponding elements of the parameter's vectors of the general θ_S and the support θ_{Rk} models.
- multiplicative superposition is used in the case of interacting processes (e.g., where some processes modify others). In this case, the outputs of the support models $f_{\theta Sv}(D_{Sv})$ are multiplied:

$$\theta_S^i = \prod_{v=1}^{b}\theta_{Sv}^i \tag{5}$$

- combined superposition methods use complex combinations of support models, such as weighted sums of outputs of nonlinear functions to combine the outputs of several models, the application of nonlinear functions to the outputs of individual models, etc. Such methods include determining the parameter vector θ_S of the approximate model as the maximum value among the corresponding components of the parameter vectors of the support models θ_{Sv}:

$$\theta_S^i = max(\theta_{Sv}^i), v = \overline{1,b} \quad (6)$$

Thus, the advantage of forming an approximate model using the support models method is the absence of a training procedure, which reduces computational complexity and significantly speeds up the process of building an approximate model. At the same time, the dimension of the approximate model $f_{\theta S}$ (the dimension of the parameter vector θ_S) remains the same as in the base models, i.e., the complexity of the approximate model does not increase. Another advantage of forming a general model using expressions (4)-(6) is the absence of a training procedure, which significantly speeds up the process of constructing a approximate model.

The algorithm for synthesizing a general model based on the superposition of support models is given below.

Algorithm 2: *general_model_sp*

1: *Input*: V, H, P, $f_{\theta Si}(D_{Si})$, D_{Si}, M, K
2: *Output*: $f_{\theta S}(D_S)$
3: *foreach* V as *Vi* :$f_{\theta Si}(D_{Si})$
4: $D_{Si} \leftarrow$ *dataset_formation*(*Vi*, H, P, D_{Si})
5: $f_{\theta Si}(D_{Si}) \leftarrow$ *init*(M, K)
5: $f_{\theta Si}(D_{Si}) \leftarrow$ *train*[$f_{\theta Si}(D_{Si})$, *time_delay*(D_{Si})]
6: *end foreach*
7: *for i* =1, ..., *v do*
8: $f_{\theta S}(D_S) \leftarrow$ *sp*[$f_{\theta Si}(D_{Si})$]
9: *end for*

Here, the *init* function initializes the model structure, *train* function learns the model, *time_delay* function forms a dataset with time delays, *sp* function perform a superposition of support models according to one of expressions (4)–(6).

4. Experiment setup

The study of the support models method is carried out on the example of a nonlinear dynamics test object. A simulation model of a test object in the form of a sequence of a nonlinear link with saturation and a dynamic link of the first order [23, 24] is shown in Fig. 2.

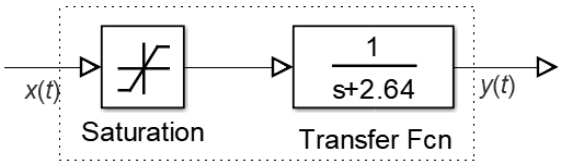

Figure 2: Test object simulation model

As typical characteristics from the set of properties of the domain of nonlinear dynamics, a nonlinear characteristic in the form of saturation and a dynamic link of the first order were chosen to describe the behavior of the test object. For the test object, a labeled D_S dataset generated on the base of signals $x(t)$ at the input of the object and the responses $y(t)$ at its output. The inputs are pulsed $x(t)=a\delta(t)$, stepped $x(t)=a\Theta(t)$, linear $x(t)=at$ and harmonic $x(t)=a\cdot sin(t)$ signals of different amplitudes $a \in (0, 1)$.

Based on the dataset D_S, two support datasets are formed:

- affiliation mark: a superscript number following the author's last name.
- input stepped signals $x(t)=a\Theta(t)$ and responses $y(t)$ of the object with nonlinearity in the form of saturation D_{R1};

- input stepped signals $x(t)=a\Theta(t)$ and responses $y(t)$ of the object in the form of a dynamic link of the first order D_{R2}.

The experiment consists in studying the training speed of the target model of the test object, built by various methods:

- affiliation mark: a superscript number following the author's last name.
- based on the general model $f_{\theta S}(D_S)$, pre-trained on the general D_S dataset;
- based on individual support models $f_{\theta R1}(D_{R1})$, $f_{\theta R2}(D_{R2})$, pre-trained on datasets D_{R1} and D_{R2};
- based on the general model $f_{\theta R}(f_{\theta R1}, f_{\theta R2})$ in the form of a superposition of support models $f_{\theta R1}(D_{R1})$ and $f_{\theta R2}(D_{R2})$.

5. Simulation and results

The structure of the target model $f_{\theta T}$ is chosen to be identical to the model based on the pre-trained general model $f_{\theta S}(D_S)$, the support models $f_{\theta R1}(D_{R1})$, $f_{\theta R2}(D_{R2})$ and the superposition of the support models $f_{\theta R}(f_{\theta R1}, f_{\theta R2})$ and is a three-layer TDNN.

According to the results of additional research, the number of neurons M=30 in the input and K=30 in the hidden layers was adopted.

The model is trained by the method of backpropagation of the error with updating the network parameters by the Levenberg-Marquardt method. Pre-training is limited to 50 epochs to prevent overtraining and preserve the ability to adapt.

Fig. 3 shows the dependencies of *mse* loss functions on the number of training epochs for target models built on the basis of the general model $f_{\theta S}(D_S)$, support models $f_{\theta R1}(D_{R1})$, $f_{\theta R2}(D_{R2})$ and superposition of support models $f_{\theta R}(f_{\theta R1}, f_{\theta R2})$.

Results of an experiment to study the learning speed of a target model of a test object built on the basis of a general model, on the basis of separate support models, and on the basis of a superposition of support models, are presented in Table 1.

The experiment shows the advantage of using support NN at the modelling nonlinear dynamics, namely, a significant reduction in the training time of the TDNN model (by 4.6 times) compared to the pre-trained general model with comparable accuracy of both models.

Applying separate support models as general ones can also reduce the training time of an accurate model (by 1.8 times).

Table 1
Results of an experiment to study the learning speed of a target model of a test object

No	Model based on	Trainong			Validation			Number of epoch
		mse	*mae*	*h*	*mse*	*mae*	*h*	
1	general model	4,9105	1,8890	1,4280	6,9738	2,3972	1,9597	14
		2,2267	1,2686	0,8311	2,9774	1,4809	1,0758	50
2	support model	4,9596	1,8623	1,3905	6,8963	2,4457	1,9411	8
	(saturation)	2,7553	1,4094	0,9604	4,1863	1,7621	1,3362	50
3	support model (first-	4,8058	1,8544	1,3913	6,8876	2,2571	1,8248	23
	order dynamic link)	3,9412	1,6869	1,2271	5,7969	2,0801	1,6669	50
4	superposition of	4,5837	1,8347	1,3811	6,6431	2,2256	1,8147	3
	support models	0,9254	0,7941	0,4115	1,0704	0,8807	0,4636	50

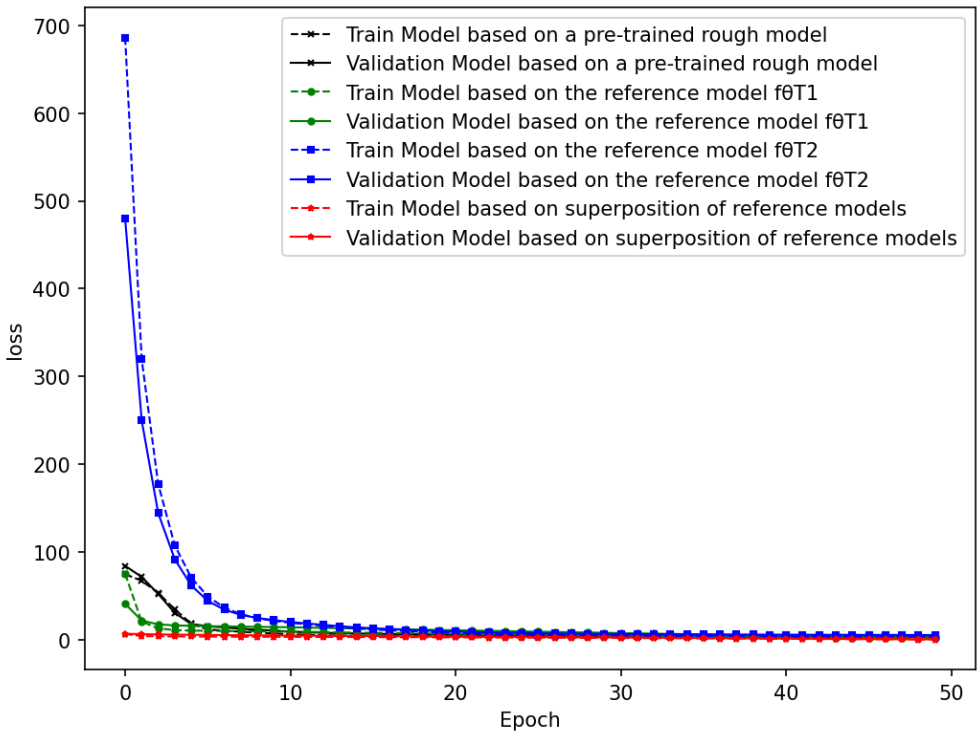

Figure 3: Dependencies of *mse* loss functions during training of target models based on the general model, based on individual support models, based on the superposition of support models on the number of learning epochs

6. Conclusion

The paper successfully solves the problem of reducing the time of building nonlinear dynamics continuous-time models in the form of neural networks while ensuring the specified accuracy of modeling. To resolve the conflict between the accuracy of modeling nonlinear dynamic objects and the speed of model construction, a modeling method was developed based on pre-training through the superposition of support models that reflect the basic properties of the subject area.

The effectiveness of the developed method for modeling nonlinear dynamics was proven when solving the problem of modelling a test nonlinear dynamic object. The experiment demonstrates a 4.6-fold reduction in the time of building a target model using support models compared to the traditional modeling method based on pre-training. The advantages of the proposed approach are the ability to quickly adapt to changing operating conditions, high speed of building the target model while ensuring the specified modeling accuracy. In addition, the developed method allows improving the efficiency of model training in the lack of labeled data for the target task. The disadvantages of the proposed approach, inherited from methods based on pre-training, are the dependence of the modeling results on the quantity and quality of data of the target dataset.

The practical limitations of the application of the proposed method are the a priori need for support models built on a sufficient amount of qualitative data. Insufficient data or poor data quality can significantly reduce the accuracy of support models and, as a result, significantly reduce the training time of an accurate model.

Thus, the area of effective application of the proposed method is allocated: lack of marked data of the target task in the presence of a general dataset of sufficient size; no significant discrepancies between the characteristics of the general and target datasets.

To improve and expand the scope of application of the support models method, it is necessary to take into account the real conditions of the external environment by expanding the experimental part at different levels of noise distortion and under conditions of time drift of the observed parameters. In order to fully assess the potential of the proposed method and determine its place

among advanced solutions in further research, it is planned to expand the range of test objects, including systems with different dynamics, multidimensional systems, objects with delays, etc.

Declaration on Generative AI

During the preparation of this paper, the authors used Google Gemini to check the coherence of the text of the article and to identify technical inaccuracies. The authors are solely responsible for the content of the publication.

References

[1] N. M. A. Chisty, H. P. Adusumalli, Applications of artificial intelligence in quality assurance and assurance of productivity, ABC Journal of Advanced Research. 11 1 (2022): 23–32. doi: 10.18034/abcjar.v11i1.625.

[2] J. Sen, Machine learning – algorithms, models and applications, IntechOpen. London: United Kingdom. 2021. doi: 10.5772/intechopen.94615.

[3] K. T. Chitty-Venkata, M. Emani, V. Vishwanath and A. K. Somani, Neural Architecture Search Benchmarks: Insights and Survey, in: IEEE Access, 2023, vol. 11, pp. 25217-25236, doi: 10.1109/ACCESS.2023.3253818.

[4] M. Li, The Past Decade and Future of the Cross Application of Artificial Intelligence and Decision Support System, in: IEEE 6th International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT), Shenzhen, China, 2025, pp. 1666-1669, doi: 10.1109/AINIT65432.2025.11036058.

[5] I. Adikari, S. Lakmali, P. Dhananjaya and D. Herath, Accuracy Comparison between Recurrent Neural Networks and Statistical Methods for Temperature Forecasting, in: 2023 3rd International Conference on Advanced Research in Computing (ICARC), Belihuloya, Sri Lanka, 2023, pp. 72-77, doi: 10.1109/ICARC57651.2023.10145620.

[6] E. Kariri, H. Louati, A. Louati, F. Masmoudi, Exploring the Advancements and Future Research Directions of Artificial Neural Networks: A Text Mining Approach. Appl. Sci. 13 (2023) 3186. doi: 10.3390/app13053186.

[7] N. K. Sinha, M. M. Gupta and D. H. Rao, Dynamic neural networks: an overview, in: Proceedings of IEEE International Conference on Industrial Technology 2000 (IEEE Cat. No.00TH8482), Goa, India, 2000, pp. 491-496 vol.2, doi: 10.1109/ICIT.2000.854201.

[8] G. Kutyniok, The mathematics of artificial intelligence, in: Proc. Int. Cong. Math. 2022; Vol. 7: 5118–5139. doi: 10.4171/ICM2022/141.

[9] Kästner, C. & Kang, E., Teaching software engineering for AI-enabled systems, The 42nd International Conference on Software Engineering (ICSE). Software Engineering Education and Training. 2020. URL: https://arxiv.org/abs/2001.06691.

[10] Hosna, A., Merry, E., Gyalmo, J. et al., Transfer learning: a friendly introduction, J Big Data. 2022; 9, (1): 102, doi: 10.1186/s40537-022-00652-w.

[11] Gholizade, M., Soltanizadeh, H., Rahmanimanesh, M. et al., A review of recent advances and strategies in transfer learning, Int J Syst Assur Eng Manag 16, 2025, 1123–1162. doi: 10.1007/s13198-024-02684-2.

[12] F. Zhuang, Z. Qi, K. Duan et al., A Comprehensive Survey on Transfer Learning, in: Proceedings of the IEEE. 99, 2020, pp. 1-34. doi: 10.1109/JPROC.2020.3004555.

[13] I. Jones, J. Swan, J. Giansiracusa, Algebraic Dynamical Systems in Machine Learning, Appl Categor Struct 32 4 (2024). doi: 10.1007/s10485-023-09762-9.

[14] W. Xu, J. He, Y. Shu, Transfer Learning and Deep Domain Adaptation, Advances and Applications in Deep Learning. IntechOpen, Dec. 09 (2020). doi: 10.5772/intechopen.94072.

[15] Y. Lu, X. Jiang, Y.Fang, Ch. Shi, Learning to Pre-train Graph Neural Networks. In: Proceedings of the AAAI Conference on Artificial Intelligence. 33, 5, 2021, 10 p. doi: 10.1609/aaai.v35i5.16552.

[16] J. Siebert, L.Joeckel, J. Heidrich, et al., Construction of a quality model for machine learning systems, Software Qual J. 30 (2022) 307–335. doi: 10.1007/s11219-021-09557-y.
[17] P. Karampiperis, N. Manouselis, T. B. Trafalis, Architecture selection for neural networks, in: Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02. Honolulu: HI, USA, 2, 2002, pp. 1115–1119. doi: 10.1109/IJCNN.2002.1007650.
[18] K. Pal, B. V. Patel, Data classification with K-fold cross validation and holdout accuracy estimation methods with 5 different machine learning techniques, in: Fourth International Conference on Computing Methodologies and Communication (ICCMC). Erode: India. 2020, p. 83–87. doi: 10.1109/ICCMC48092.2020.ICCMC-00016.
[19] K. Nakamichi, et al., Requirements-driven method to determine quality characteristics and measurements for machine learning software and it's evaluation, in: IEEE 28th International Requirements Engineering Conference (RE), Zurich, Switzerland, 2020. p. 260–270, doi: 10.1109/RE48521.2020.00036.
[20] T. Warren Liao, Clustering of time series data—a survey, Pattern Recognition, 38 11 (2005) 1857-1874. doi: 10.1016/j.patcog.2005.01.025.
[21] G. Leylaz, S. Wang, JQ Sun, Identification of nonlinear dynamical systems with time delay, Int. J. Dynam. Control. 2022. 10, 13–24. doi: 10.1007/s40435-021-00783-7.
[22] L. Wenyuan, L. Zhu, F. Feng, at al., A time delay neural network based technique for nonlinear microwave device modelling, Micromachines. Basel, 11 9 (2020) 831. doi: 10.3390/mi11090831.
[23] O. Fomin, et al. 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. Lecture Notes in Networks and Systems. Springer. Cham. 536 (2022) 258–265. doi: 10.1007/978-3-031-20141-7_24.
[24] O.O. Fomin, A.A. Orlov, Modeling nonlinear dynamic objects using pre-trained time delay neural networks. Applied Aspects of Information Technology. 7 1 (2024) 24–33. doi: 10.15276/aait.07.2024.2.