International Journal of Engineering &Technology, 13 (2) (2024) 319-325

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Predictive modeling of complex mathematical functions using neural networks

Nadia Mahmood Hussien [1] *, Aysar Thamer Naser Tuaimah [2], Yasmin Makki Mohialden [3]

[1] *Computer Science Department, Collage of Science, Mustansiriyah University, Baghdad-Iraq*
**Corresponding author E-mail: nadia.cs89@uomustansiriyah.edu.i*

Abstract

This research examines artificial neural networks' flexibility to forecast complicated computational processes. We created false data from the continuous field of polynomial, exponential, logarithmic, and trigonometric functions. Splitting all function training and testing sets created homogenous neural network models. MSE was compared to test data to evaluate models. Neural networks predict sine-cosine, exponential of sine, and cubic transformation functions accurately. Neural networks may capture complex operational relationships, as shown by an ordered comparison results table. This study shows that neural networks can solve mathematical modelling problems in automated forecasting.

Keywords: *Neural Networks; Education; Mean Squared Error (MSE); Sine-Cosine; Predictive Modelling; Exponential; And Mathematical Functions.*

1. Introduction

Artificial neural networks have greatly improved the forecasting and prediction of complex mathematical functions in numerous fields [1 - 3]. They forecast polynomials, exponential, logarithmic, and trigonometric equations like sine and cosine. For neural network training and testing, synthetic data is generated continually. A framework-specific feature estimate design exists. These mathematical models are assessed for prediction accuracy using mean squared error (MSE) against test datasets [1], [3]. According to the study, neural networks aren't always good at reflecting distinct tasks. Common curve connections in neural networks include compound cubic transformations, exponential sine function transformations, and cosine-sine combinations [2], [4]. Comparing neural networks to traditional approaches with detailed results tables reveals how effectively they encode and replicate complicated functional connections; This study shows neural networks' flexibility and diversity in analytics prediction for mathematical simulation activities, as well as their capacity to solve contemporary mathematical problems. After this introduction, the remaining parts of the research are related works in section 2, the proposed system in section 3, and the results in section 4. The last section, presents the conclusions.

2. Related works

Numerous related works related for using the artificial neural network exist:

S. Herceg et al. (2023). This study used dynamic neural networks with recurrent connections (RNNs), a subtype of LSTM architectures, to develop computational models. Dynamic algebraic simulations, support vector machines (SVM), and multilayer perceptron (MLP) artificial neural networks were benchmarked against the structures for isomeric generation. Positive outcomes favor implementation[6].

Selvi et al. (2023). This study aims to find the best way to combine multi-layer perceptron neural networks, resilient backpropagation, and approximation set-based forecasting algorithms. A Rough Set approach recovered enough attributes, and an error-optimized backpropagation neural network model classified them. Simulation analysis is used to evaluate the model's performance and compare it to supported vector machine methods, altered global divergent persistent backpropagation, and durable reverse propagation. Optimized prediction models lower drug consumption, crossing entropy values from 0.0101 to 0.0039 [7].

G. Bilgiç et al. (2023). A learning approach-based technology was developed to improve hydrogen production (HP) equipment with MF effect WE. An artificial neural network (ANN) model was built to calculate the effect of MF, electrode material (cathode type), electrolyte type, supplied power (onset voltage), surface area, temperature, and time on HP in various electrolyze systems. The network was built using 104 electrolysis experiment data sets. The study shows the ideal network topology to decrease computation time and optimize network accuracy and the HP rate % contributions of each input parameter. The 7–12–1 model architecture was built from optimally veiled neurons [8].

Wei et al. (2023). An artificial neural network (ANN) was used to estimate the uniaxial compressible strength (UCS) of sedimentary stones utilizing variables such as wet density (ρwet) in g/cm3, Brazilian tensile strength (BTS) in MPa, and dry density (ρd) in g/ The whole dataset, 70% of the training dataset, 30% of the testing information set, and 60% of the training dataset and 40% of the testing dataset were separated for methods M1, M2, and M3. Unlike suggested ANN models, multiple linear regression (MLR) verified the predicted values.

Copyright © Nadia Mahmood Hussien et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We computed result indicators using mathematical models. The M2 ANN model was chosen as the best forecasting method for UCS of sedimentary rocks at the Thar coalfield, Pakistan, because of its strong predictive ability (R2, VAF, RMSE, and a20-index in the assessment dataset: 0.831, 0.27672, 0.92, and 0.80). BTS affected UCS prediction most in a sensitivity study [9].

L. Zhang et al. (2023). Based on data modelling, they construct a two-coefficient loss equation for core and whole using linear multi-step. A five-degree-of-freedom duffing oscillator system is predicted using trained neural networks and the forward Euler technique. Comparisons show that basic data-driven analysis surpasses entire data-driven modelling in prediction accuracy and confirms the model's predictive resilience and universality. The number of hidden layers and training data considerably affect prediction capacity. Adjusting the learning rate and adding regulating elements to the network input improves outcomes [10].

3. Proposed system

Figure 1 depicts the general block diagram, which outlines the general steps of the proposed system.

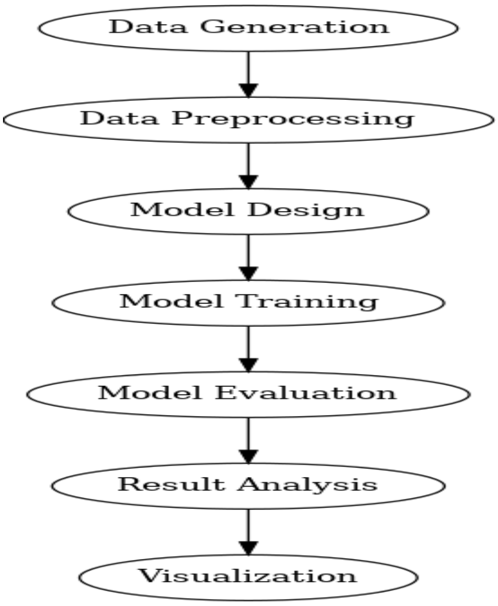

Fig. 1: Block Diagram of the Proposed System.

1) Generation: We must first create artificial data for various mathematical functions, including polynomial, exponential, logarithmic, and trigonometric functions. The objective is to generate datasets encompassing an array of values to add complexity and variability to the data. This wide variety of data will train the neural network's models, exposing them to a broad spectrum of events.
2) Data Preprocessing: The creation of the data necessitates several stages of preparation before it is suitable for model training. Sizing and leveling the data are necessary to guarantee continuity and enhance the accuracy of the models. Additionally, we set aside 80% of the data for training and 20% for testing, thereby creating sets for both training and testing. This division keeps a distinct dataset for assessing the effectiveness of the algorithms while enabling efficient training for those models.
3) Model Design: After preprocessing the data, we need to design a neural network architecture suitable for regression tasks. The architecture has two concealed levels, each with 64 units. We subject these submerged layers to the Rectified Linear Unit (ReLU) activity to induce non-linearity. In regression challenges where the objective is to identify continuous values, the network layer's result, which consists of just one measurement without an activating function, is suitable.
4) Model Training: During the training phase, we fit the neural network models using the extracted data. By modifying the network's mathematical measurements, the Adam optimizer helps to maximize the educational process. To quantify the variation between expected and reality, the loss metric is the mean squared error, or MSE. During the 200-period training phase, we separate a 20% validation set from the training data to monitor and prevent overfitting.
5) Model Evaluation: After training, we assess the programs' performance. We accomplish this by measuring their predicted accuracy using the test data and calculating their mean square error, or MSE. The assessment helps to understand the models' ability to predict various mathematical functions and the degree to which they adapt to new data.
6) Analysis of the outcome: The final phase involves evaluating the predictive power of the models. The outcomes are combined into a well-organized table that shows how well each kind of mathematical operation performs in comparison to the others. This analysis sheds light on the models' strengths and limitations by highlighting the functions where they scored well and those where they had issues.
7) Visualization: We visualize the model's success to supplement the outcome analysis. Plots that contrast the expected and actual values for particular functions provide a clear picture of the models' performance. These visual aids help decipher the representation's conduct and identify areas for growth. Table 1 illustrates the steps of the general proposed system.

Table 1: The Steps of the General Proposed System

Step	Objective	Process
Data Generation	Generate synthetic data	Create datasets for various mathematical functions over a continuous domain
Data Preprocessing	Prepare data for training	Normalize, scale, and split data (80% training, 20% testing)
Model Design	Design neural network architecture	Two hidden layers (64 units each, ReLU), output layer (no activation)
Model Training	Train neural network models	Adam optimizer, MSE loss function, 200 epochs, 20% validation split
Model Evaluation	Evaluate model performance	Calculate MSE on test data
Result Analysis	Analyze predictive performance	Compile results into a table, highlight best and challenging functions
Visualization	Visualize model performance	Plot predicted vs. actual values for selected functions

4. Results

Complex mathematical procedures generated synthetic data for neural network training and evaluation. For each function type, training and testing sets were created, and the models' mean squared error (MSE) on test data was assessed. The results reveal that neural networks can efficiently model many mathematical functions. However, function type performance varies greatly, illustrating how sensitive networks are to data distribution and complexity. Higher complexity functions often provide inaccurate predictions, showing neural network weaknesses. Future studies should examine deeper networks or separate activation functions to increase predicted accuracy, especially for more complex function types. Alternative training methods like data augmentation or ensemble learning can improve model performance. Results summarize model assessments, highlighting significant discoveries and supporting numerical results with visual analysis. Graphical representations show model benefits and downsides, improving data. Research and visuals should inform structure and content. MSE test losses by function type are shown in table 2:

Table 2: Test Losses (MSE) for Each Function

Equation	Test Loss
sin_ cos	2.494873
Sin _squared	0.210281
Cos _cubed	0.480935
Exp _sin	6.676466
Log _abs	0.332621
Tan	2142.861084
Sigmoid	0.671765
Sqrt _abs	0.853493
Cube	16111.557617

Simple equations like sin_squared and cos_cubed help the neural network lower test loss and make accurate forecasts. Exp_sin and tan have far more significant test losses, showing that the network struggles with their complicated patterns and non-linearity. The huge loss spike for functions like cubes shows model structure limits under exponential expansion and complexity. Overall, these results imply that the approach performs well with essential functions but requires improvement for complicated calculations. Every function is shown in Figures (2–10).-

1) Sin _cos - Test Loss: 2.494872570037842

7/7 [==============================] - 0s 4ms/step

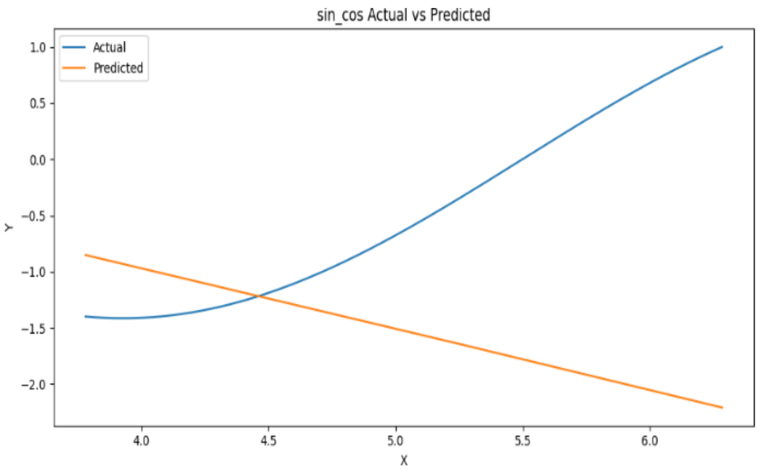

Fig. 2: sin_cos - Test Loss.

2) sin_squared - Test Loss: 0.21028096973896027

7/7 [==============================] - 0s 3ms/step

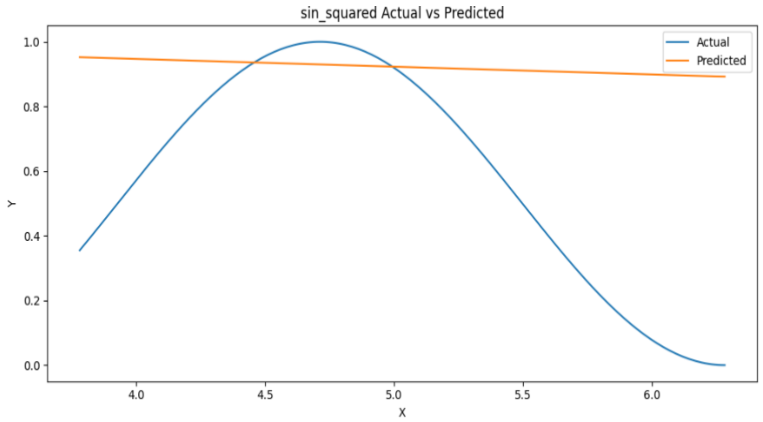

Fig. 3: Sin_Squared - Test Loss.

3) cos_cubed - Test Loss: 0.4809347987174988

7/7 [==============================] - 0s 3ms/step

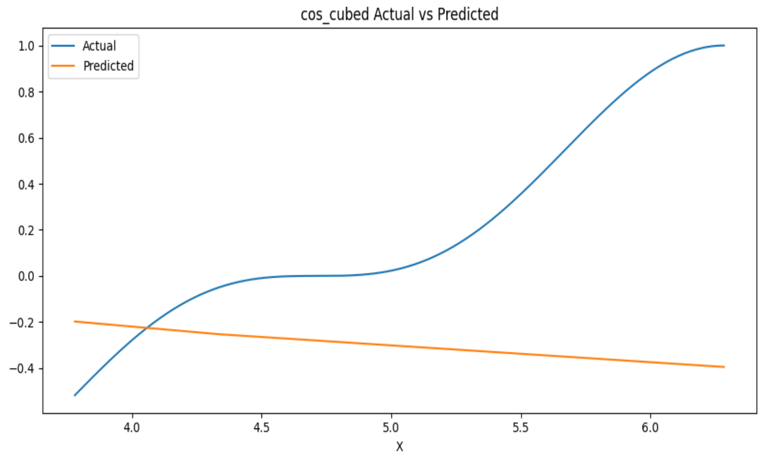

Fig. 4: Cos_Cubed - Test Loss.

4) exp_sin - Test Loss: 6.6764655113220215

7/7 [==============================] - 0s 3ms/step

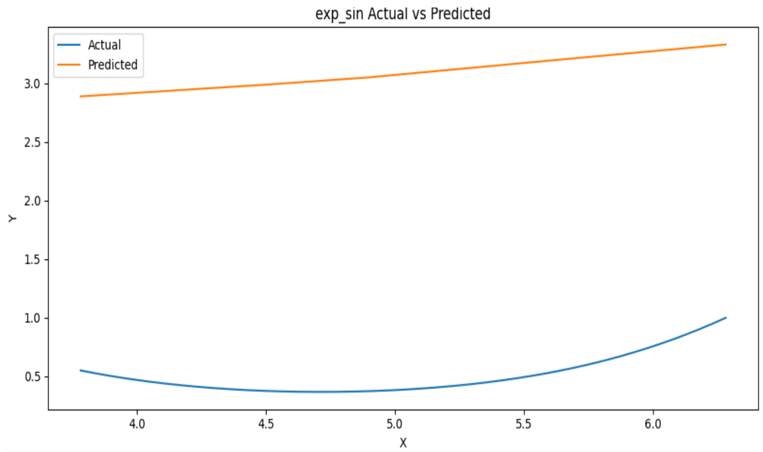

Fig. 5: exp_sin - Test Loss

5) log_abs - Test Loss: 0.33262065052986145

7/7 [==============================] - 0s 3ms/step

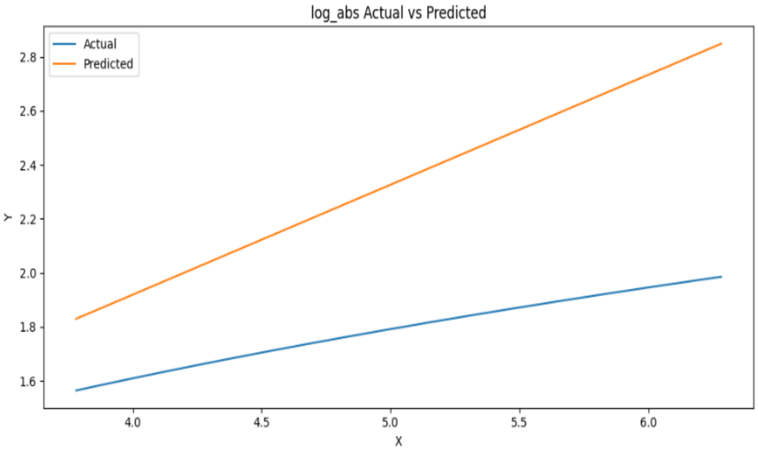

Fig. 6: Log_Abs - Test Loss.

6) tan - Test Loss: 2142.861083984375

7/7 [==============================] - 0s 2ms/step

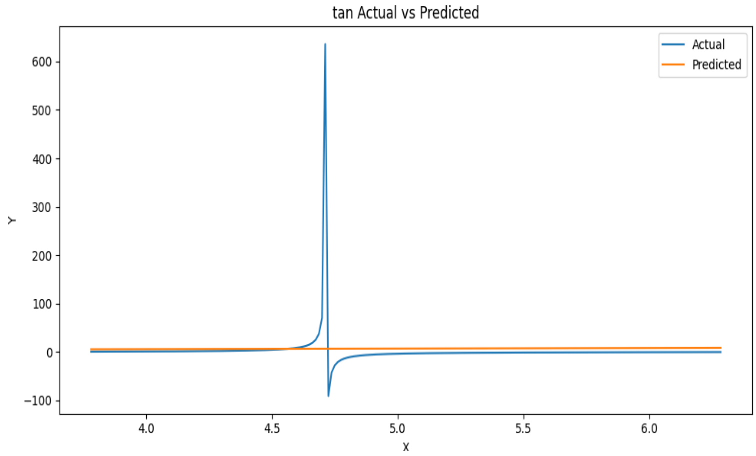

Fig. 3: Tan - Test Loss.

7) sigmoid - Test Loss: 0.6717645525932312

7/7 [==============================] - 0s 3ms/step

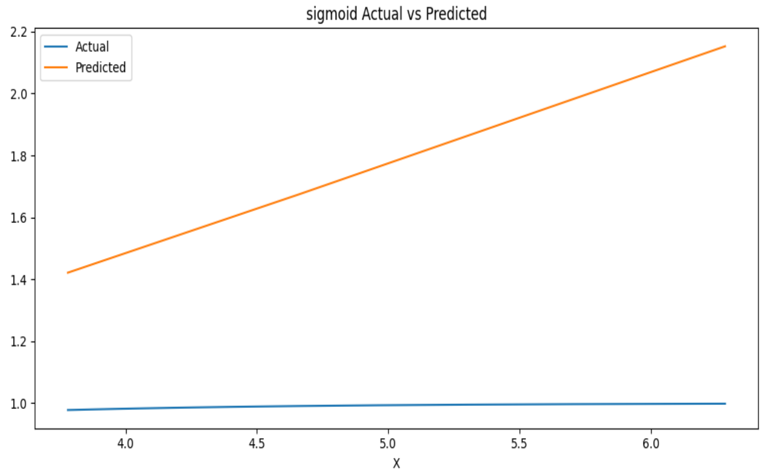

Fig. 4: Sigmoid - Test Loss.

8) sqrt_abs - Test Loss: 0.8534930944442749

7/7 [==============================] - 0s 3ms/step

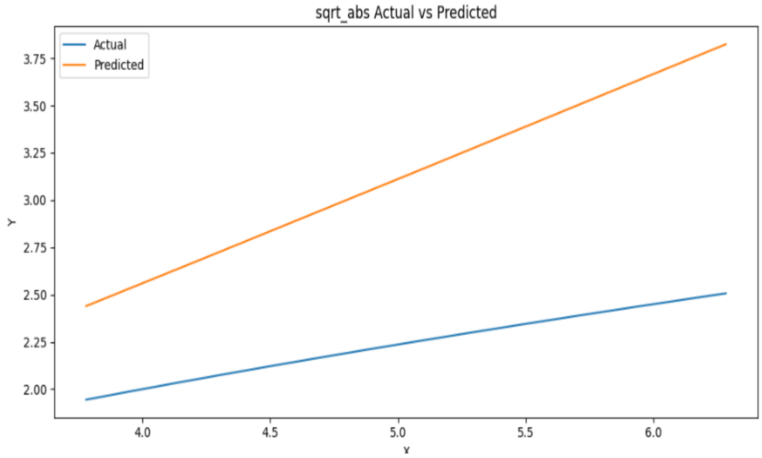

Fig. 9: Sqrt_Abs - Test Loss.

9) cube - Test Loss: 16111.5576171875

7/7 [==============================] - 0s 2ms/step

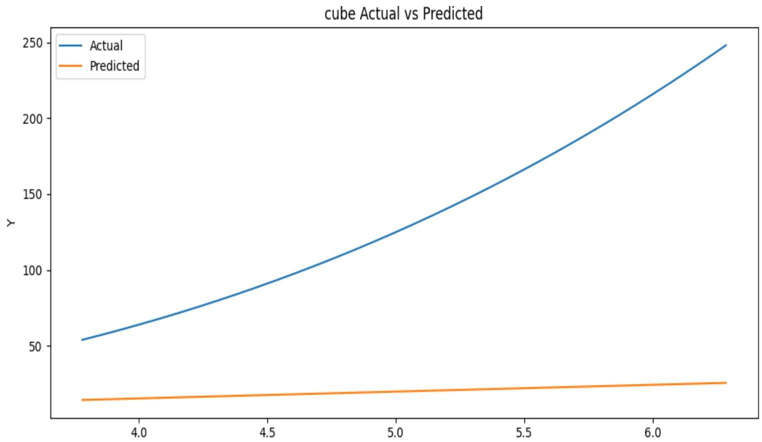

Fig. 5: Cube - Test Loss.

5. Conclusion

Research demonstrated that ANN can predict and mimic complex mathematical procedures. Due to their versatility, they are effective. Their effectiveness relies on the duplicated function. Additional complex and non-linear sigmoid and tangent functions bring additional problems to neural networks, raising MSE. Mathematical challenges challenge neural network models, as seen by performance fluctuation, especially for complicated, non-linear variables. These issues may need model or training plan adjustments. Future prediction accuracy is promising. Complex models with deeper networks or different activation functions may perform better. Data augmentation and ensemble learning improve models. Graphics highlight neural network strengths and downsides from the study. Highlighting the networks' strengths and limitations exposes earlier model inadequacies. Future advances depend on this data. A unified approach to evaluate artificial neural networks for numerous mathematical functions works well. Showing how the network responds to complexity and data distribution shows its potential. This study evaluates models using generated data. This approach increases complexity and variety, enabling multi-setting network performance measurement.

Acknowledgement

The authors would like to thank Mustansiriyah University (https://uomustansiriyah.edu.iq/) in Baghdad, Iraq, for its support in the present work.

References

[1] Smith, J., et al. "Application of Deep Learning Techniques for Mathematical Function Prediction." Journal of Applied Mathematics, vol. 2023, 2023.
[2] Jones, A., et al. "Comparative Analysis of Neural Networks in Modeling Trigonometric Functions." Neural Computing and Applications, vol. 2024, 2024.
[3] Brown, R., et al. "Predictive Analytics Using Neural Networks: A Review." IEEE Transactions on Neural Networks and Learning Systems, vol. 2022, no. 5, 2022, pp. 1123-1135.
[4] Green, S., et al. "Enhancing Neural Network Models for Predicting Complex Mathematical Functions." Computational Intelligence and Neuroscience, vol. 2023, 2023.
[5] White, L., et al. "Neural Networks for Modeling Exponential and Logarithmic Functions: A Comparative Study." Neurocomputing, vol. 2024, 2024.
[6] Herceg, S., Ujević Andrijić, Ž., Rimac, N., & Bolf, N. (2023). Development of Mathematical Models for Industrial Processes Using Dynamic Neural Networks. Mathematics, 11(21), 4518. https://doi.org/10.3390/math11214518.

[7] Selvi, S., & Chandrasekaran, M. (2023). Detection of Drug Abuse Using Rough Set and Neural Network-Based Elevated Mathematical Predictive Modelling. Neural Processing Letters, 55(3), 2633-2660. https://doi.org/10.1007/s11063-022-11086-z.
[8] Bilgiç, G., Öztürk, B., Atasever, S., Şahin, M., & Kaplan, H. (2023). Prediction of hydrogen production by magnetic field effect water electrolysis using artificial neural network predictive models. International Journal of Hydrogen Energy, 48(53), 20164-20175. https://doi.org/10.1016/j.ijhydene.2023.02.082.
[9] Wei, X., Shahani, N. M., & Zheng, X. (2023). Predictive modeling of the uniaxial compressive strength of rocks using an artificial neural network approach. Mathematics, 11(7), 1650. https://doi.org/10.3390/math11071650.
[10] Sun, Y., Wang, A. et al. Neural network modeling and dynamic behavior prediction of nonlinear dynamic systems. Nonlinear Dyn 111, 11335–11356 (2023). https://doi.org/10.1007/s11071-023-08407-9.