

# Reactivity margin evaluation software for WWR-c reactor\*

Ivan P. Belyavtsev<sup>1</sup>, Sergey O. Starkov<sup>1</sup>

<sup>1</sup> Obninsk Institute for Nuclear Power Engineering National Research Nuclear University “MEPhI”, 1 Studgorodok, Obninsk, Kaluga reg., 249020 Russia

Corresponding author: Ivan P. Belyavtsev (djbelyak@gmail.com)

---

**Academic editor:** Yury Kazansky ♦ **Received** 20 August 2018 ♦ **Accepted** 16 November 2018 ♦ **Published** 7 December 2018

---

**Citation:** Belyavtsev IP, Starkov SO (2018) Reactivity margin evaluation software for WWR-c reactor. Nuclear Energy and Technology 4(3): 197–201. <https://doi.org/10.3897/nucet.4.31862>

---

## Abstract

The WWR-c reactor reactivity margin can be calculated using a precision reactor model. The precision model based on the Monte Carlo method (Kolesov et al. 2011) is not well suited for operational calculations. The article describes the work on creating a software package for preliminary evaluations of the WWR-c reactor reactivity margin.

The research has confirmed the possibility of using an artificial neural network to approximate the reactivity margin based on the reactor core condition. Computational experiments were conducted on training the artificial neural network using the precision model data and real reactor measured data. According to the results of the computational experiments, the maximum relative approximation error  $\Delta k/k$  for fuel burnup was 3.13 and 3.56%, respectively. The mean computation time was 100 ms.

The computational experiments showed it possible to construct the artificial neural network architecture. This architecture became the basis for building a software package for evaluating the WWR-c reactor reactivity margin – REST API based web-application – which has a convenient user interface for entering the core configuration. It is also possible to replenish the training sample with new measurements and train the artificial neuron network once again.

The reactivity margin evaluation software is ready to be tested by the WWR-c reactor personnel and to be used as a component of the automated reactor refueling system. With minor modifications, the software package can be used for reactors of other types.

---

## Keywords

WWR-c reactor; reactivity margin; artificial neural networks; software package; generalized approximation theorem

---

## Problem definition

A WWR-c experimental nuclear facility (heterogeneous water-water research reactor) has been in operation since 1964 at the branch of the L.Ya. Karpov Institute of Physical Chemistry (NIFHI) near the town of Obninsk, Kaluga

region. The reactor is specialized for a wide range of research works in the field of radiation chemistry, structural and material research, activation analysis, neutron transmutation doping of semiconductors, etc. (Kochnov et al. 2008). In 1980, the Institute began to produce reactor-based medical radionuclides and radiopharmaceuticals. In

\* Russian text published: Izvestiya vuzov. Yadernaya Energetika (ISSN 0204-3327), 2018, n. 2, pp. 58-66.

1986, due to the successful development of this area and its convenient geographical location, it was decided to reconstruct the reactor (Kolesov et al. 2011).

In 2011, the need to improve the reactor parameters and increase the efficiency of radionuclide ( $^{99}\text{Mo}$ ,  $^{131}\text{I}$ , etc.) production necessitated creating a precision neutron-physical calculation model of the reactor core, reflector and CPS bodies. Modeling took into complete account the geometry of all fuel rods (fuel, cladding, water gap with appropriate temperatures), the fuel isotopic composition changes depending on burnout, the geometry and composition of CPS bodies, reflectors, experimental channels, and structures. The obtained precision model was verified for reactor reactivity margin calculations (Kolesov et al. 2011).

The model is based on the Monte Carlo method, which makes it possible to achieve high accuracy in modeling the reactor core physical processes. However, this approach requires a large amount of computer time to conduct computational experiments; for example, it takes about eight hours to calculate the reactivity margin for one core configuration.

The WWR-c reactor campaign is 100 hours per week with subsequent shutdown for cooling down, fuel/target reloading and other technological operations (Kochnov et al. 2014). Due to such a short campaign, it is rather difficult to use a precision model for carrying out operational calculations: With a single computing system being fully utilized, no more than 12 calculations can be made during one campaign. Thus, there appears a task to create a software package for evaluating the WWR-c reactor reactivity margin. This software package should assist the reactor R & D personnel in carrying out preliminary calculations of the reactivity margin. The main requirement for this software is to increase manifold the computational performance while maintaining a sufficient level of accuracy. The main task of the software package is to approximate the reactor reactivity margin depending on the fuel burnout and the position of the CPS bodies.

## Using artificial neural networks to approximate the reactivity margin

In general, artificial neural networks (ANN) denote an approach to the construction of computational algorithms and devices based on their similarity to biological neurons (Simon Haykin 1999). In this work, artificial neural networks are considered as a family of information processing algorithms.

An artificial (formal) neuron is an elementary computational cell of an artificial neural network. Each artificial neuron receives a vector of input signals  $\mathbf{x} = (x_0, x_1, \dots, x_n)$ , for which a weighted sum is calculated. Then, the activation function value ( $\varphi$ ) is calculated from this weighted sum:

$$n(\mathbf{x}) = \varphi\left(\sum_{i=0}^n w_i x_i + b\right)$$

where  $\mathbf{w}$  is the weight vector and  $b$  is the offset.

A set of artificial neurons that receive a single vector of input signals is called a fully connected neural layer. A sequence of neural layers, in which the vector of output signals of the previous layer is the input vector of the next layer, is called a multilayer perceptron.

All weights in a multilayer perceptron are initialized by random small values. In this configuration, the perceptron produces noise in response to any input vector. To configure the perceptron to perform a given function, iterative training is performed. The training process consists in presenting a vector from the training data set to the neural network input, obtaining the result at the neural network output, comparing the output with the expected one, and adjusting the weights to reduce the difference obtained. One of the most frequently used training algorithms is gradient descent (Filatova 2004).

Let us justify the possibility of using an artificial neural network to construct an approximation. Approximations can be constructed using a generalized approximation theorem. According to this theorem, it is possible to obtain an arbitrarily accurate approximation of any continuous multivariable function using the operations of addition and multiplication by a number, superposition of functions, linear functions, and one arbitrary continuous non-linear one-variable function (Gorban 1998). Since these operations are fully implemented by an artificial neural network with one non-linear formal neuron, it is permissible to use an artificial neural network to construct the required approximation.

To confirm the possibility of approximating the reactivity margin using an ANN, two computational experiments were carried out. In the first experiment, an artificial neural network was trained on the data calculated using a precision model. In the second experiment, an artificial neural network was trained using real data from the WWR-c reactor campaigns.

## Model data approximation

To conduct an experiment on the model data approximation by an artificial neural network, a data set was constructed. Using a precision model, computational experiments were carried out for 34 different reactor configurations (fuel assembly burnup, CPS position) and a reactivity margin value was obtained for each configuration. Based on the obtained data, a training data set (25 configurations) and a test dataset (9 configurations) were formed. For the final verification, all 34 core configurations were used.

For the experiment, a three-layer artificial neural network was created. The input layer consists of 50 formal neurons with the ReLu activation function:

$$f(x) = \begin{cases} 0, & x < 0; \\ x, & x \geq 0. \end{cases}$$

The hidden layer consists of 10 formal neurons with the ReLu activation function. The output layer consists of one formal neuron with the logistic activation function

$$f(x) = (1 + e^{-x})^{-1}.$$

For training the artificial neural network, 50,000 training epochs were conducted. Training was based on the back-propagation mean square error (MSE) for the training data set. Every 100 epochs, the mean square error was evaluated on the test data set. Throughout the training process, the error reduced to zero without divergence.

After the training process was completed, the ANN was verified on the model data. Figure 1 shows the complete results of the comparison of precision reactivity margin calculations with the results of the ANN work. The verification results are generalized as follows:

- the mean absolute approximation error = 0.0405;
- the maximum absolute approximation error = 0.1029;
- the mean relative approximation error = 1.21%;
- the maximum relative approximation error = 3.13%;

The average time required to calculate the reactivity margin using the artificial neural network is 100 ms (Belyavtsev et al. 2018).

## Measured data approximation

To conduct the experiment on the measured data approximation, data from 24 real reactor campaigns were taken.

The data were divided into two sets: a training data set (18 campaigns) and a test data set (6 campaigns). To validate the trained ANN, data from all 24 campaigns were used.

The ANN architecture is identical to the architecture in the first experiment. For training the ANN, 50,000 training epochs were conducted. Training was also based on the back-propagation mean square error for the training sample. Every 100 epochs, the mean square error was evaluated on the test data set. Throughout the training process, the error reduced to zero without divergence.

After the training process was completed, the ANN was validated on the measured data. Figure 2 shows the complete results of the comparison of measured reactivity margin values with the results of the ANN work. The validation results are generalized as follows:

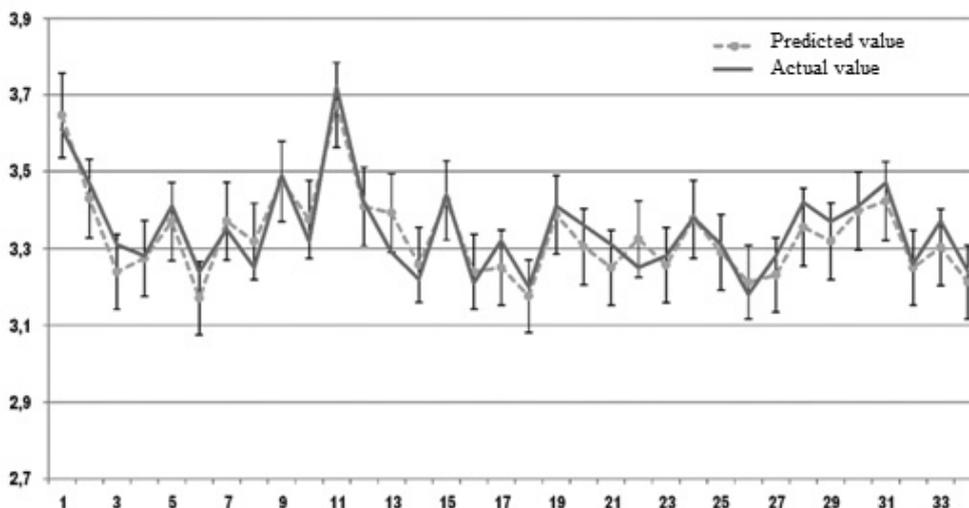
- the mean absolute approximation error = 0.0412;
- the maximum absolute approximation error = 0.1159;
- the mean relative approximation error = 1.26%;
- the maximum relative approximation error = 3.56%;

The average time required to calculate the reactivity margin using the artificial neural network is 100 ms.

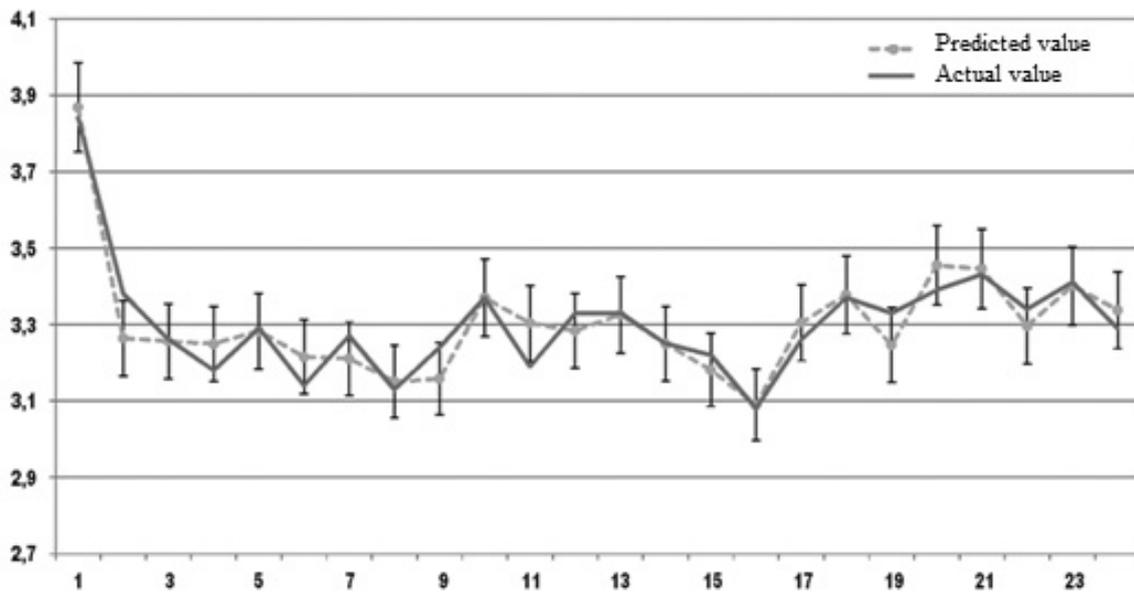
The described computational experiments showed that the obtained neural networks implement a correct approximation and have high accuracy and operating speed (Belyavtsev et al. 2018).

## Critically margin evaluation software for WWR-c reactor

Based on a finite number of precision calculations or changes, and using an artificial neural network, it became possible to implement the procedure for approximating evaluations of the WWR-c reactor reactivity margin in a software package. For reactor core configurations within



**Figure 1.** Performance verification graph of an artificial neural network trained on model data (measurement numbers in the variational sample are plotted on the abscissa; the reactivity margin values, %  $\Delta k/k$ , are laid off on the ordinate)



**Figure 2.** Performance validation graph of an artificial neural network trained on measured data (measurement numbers in the variational sample are plotted on the abscissa; the reactivity margin values, %  $\Delta k/k$ , are laid of on the ordinate)

the training sample, a quick and fairly accurate evaluation of the reactivity margin can be obtained.

The next step is to ensure the possibility of using artificial neural networks for preliminary reactivity margin calculations. To solve this problem, it was necessary to create a software package for evaluating the WWR-c reactor reactivity margin.

This software package must meet the following requirements:

- to replenish the training sample for an artificial neural network;
- to work in training modes and use a trained neural network;
- to have a user-friendly and intuitively comprehensible interface;
- to be easy-to-install and easy-to-use.

The developed software package consists of the following structural elements:

- artificial neural network;
- training data storage;
- REST API for data exchange;
- user interface.

Let us consider these elements.

The basic component for creating an artificial neural network is the TensorFlow framework (Abadi et al. 2015, Zaccane 2016, Lieder et al. 2017), which builds and executes a graph of computations in heterogeneous computing systems, has a rich library of primitives for constructing artificial neural networks, and ensures the effective use of available computing environments.

The DNNRegressor primitive from the TensorFlow library was used to create an artificial neural network. The network input layer contains 50 formal neurons with the ReLU activation function, the hidden layer is composed of 10 formal neurons with the ReLU activation functions,

and the output layer is one formal neuron with the logistic activation function. Thus, this structure completely repeats the network architecture used in the computational experiments.

The DNNRegressor primitive can function in training, evaluating, and predicting modes.

The DNNRegressor uses training and evaluating modes to train and predicting mode to use the ANN. The pre-trained DNNRegressor primitive is stored in a tf.data file, which contains a description of the prediction graph (Inference) and all the weights of the formal neurons.

Comma-Separated Values (CSV) files are used to store training data. Each entry is a string with sequentially recorded burnout percentages of each fuel assembly, CPS positions, and reactivity margin values for given core configurations. For the training procedure, data from CSV files are brought into memory in the form of NumPy arrays.

To ensure interaction with users, a web-based control interface (You 2018, Filipova 2016, Street 2017) was implemented using the VueJs library: this interface is a simplified reactor map. It is possible to specify the burnout percentage for each fuel assembly and the position for each CPS body (see Fig. 3). Depending on the selected operating mode, the reactor criticality in a given configuration will be estimated or another value will be added to the training set. To provide communication between the software parts, using the Flask library (Ronacher 2018, Grinberg 2018, Dwyer 2016), a representational state transfer (REST) application programming interface was implemented (Masse 2011, Richardson and Ruby 2008).

## Conclusions and prospects

