STUDIA INFORMATICA
Nr 2 (31) Systems and information technology 2024

Dariusz Ruciński[1]
ORCID: 0000-0001-5458-9170

University of Siedlce
Faculty of Exact and Natural Sciences
Institute of Computer Science
ul. 3 Maja 54, 08-110 Siedlce, Poland

[1]dariusz.rucinski@uws.edu.pl

Selected Aspects of Neural-Evolutionary Modeling of Prices on the Day Ahead Market of TGE S.A.

DOI: 10.34739/si.2024.31.06

Abstract. Modeling in the context of Artificial Intelligence (AI) is using mathematics to describe, analyze, and predict real-world systems. Building models that can simulate or predict various aspects of reality is a key issue that is the subject of many studies. The quality of models depends on many elements, starting from the architecture of the neural network itself, through the selection of teaching data in terms of the size of the sets, and the number of factors influencing the choice of the network itself. Modifications of the network training methods themselves also play an important role, e.g. through the use of Evolutionary Algorithms (AE). The paper focuses on several selected aspects related to the quality of modeling based on prices on the Day Ahead Market (DAM). The influence of network architecture factors, network type, number of training data, and Evolutionary Algorithms on the improvement of the model quality measured by the Mean Squared Error (MSE) and the coefficient of determination (R^2) were considered.

Keywords: Day-Ahead Market, Modeling, Neural Networks, Evolutionary Algorithms, model improvement, MATLAB environment.

1 Introduction

During the development of modeling, several different approaches have been developed and have introduced their own research tools. These include statistical models that use data to predict future outcomes based on probability distributions. optimization models such as linear programming, which optimizes a linear objective function under linear constraints, or Evolutionary Algorithms for optimizing problems by simulating evolutionary processes. The so-called machine learning includes neural networks that are trained based on input data. The work uses a model with a teacher where the model output data is compared with the system output data for prognostic purposes, this is the so-called regression model. There are many types of neural networks, from the historically first Perceptron [19] through Recurrent Neural Networks (RNN) [8], Radial Basis Function Networks (RBF) [18], to deep networks, i.e. Long Short-Term Memory (LSTM) networks [6] or Convolutional Neural Networks (CNN) [13, 25]. Both the diversity of network types and the search for different solutions within a given type of network, e.g. the number of layers, learning algorithm, etc., serve one purpose, namely building a model of the real system for various purposes, e.g. optimization, prediction, or simulation. The subject of this work is the presentation of selected results of work related to the modeling of the Day-Ahead Market (DAM) system of the Electricity Exchange in Poland.

2 Day Ahead Market

The basic principles of operation of the electricity exchange (ee) system are included in the Energy Law Act of April 10, 1997 and related implementing acts. The Energy Law does not provide for specific restrictions on shaping various methods of energy trading. Currently, the Polish electrical energy (ee) system consists of three subsystems (called segments): the contract market system, the stock market system and the balancing market system [16].

The contract market system is a system that trades electricity based on bilateral contracts concluded directly between electricity producers and final customers or companies trading in electricity. The exchange market system is a system that trades energy on the energy exchange (TGE S.A.), mainly on the DAM (the Intraday Market also operates on TGE S.A.). Listed on TGE S.A. futures contracts for the supply of electricity are used to determine the price in the long term (in the so-called long-time horizon), which is used by sellers and large consumers to forecast the price of electricity and optimize the costs of its sale and purchase. TGE S.A. also trades in property rights arising from certificates of origin of energy, where producers of energy from renewable energy sources and cogeneration sources and companies obliged to purchase certificates of origin can trade in property rights to these certificates. At TGE S.A. CO_2 emission allowances are also traded when European Unit Allowance (EUA) units are traded in the form of spot transactions. Quotations on the DAM taking place every day in two sessions: at 8:00 and 10:30 held the day before the day on which the physical delivery of electrical energy takes place. The system consists of 24-hour quotation settlement periods in which members submit purchase and sale offers, the price is determined on market principles on the so-called fixings, there are also continuous quotations.

The overriding challenge is also the need to integrate the IT management system of TGE S.A. with the National Power System in order to include DAM system into a unified Electricity

Market system, and therefore there is a need to obtain a model of the DAM system to include it online in the above-mentioned integrated system.

There are many approaches to modeling prices in this market in the literature. The most popular method is to use time series of historical energy prices, which are fed as input to Artificial Neurons Network (ANNs). However, some studies also take into account additional factors at the ANN input, such as the volume of energy sold, which allows for a more comprehensive analysis. This work proposes a systemic approach using Multiple Input Multiple Output (MIMO) models to create price models. Key element of proposed method is to determine in advance the volume of energy that serves as input in order to be able to forecast weighted average energy prices as the final result.

The proposed model assumes that the energy volume is shifted by one day in relation to the quoted price. Other studies based on historical energy prices also use a similar offset approach. There are also analyzes in the literature that use energy volumes as input instead of shifted historical prices. Furthermore, it is worth noting that there are methods for forecasting both energy volumes and power demand that can be used as inputs in the models presented in this work.

3 Model description

In terms of the possibility of using ANN as a regression model, the literature on the subject distinguishes at least the following ways of using data to obtain a model:

- Price time series using a creeping trend model, in which volume-weighted average energy prices with a daily shift (t+1) are assumed as input values [17],

- Time series with the following input values: energy price (t), volume (t), wind speed (t), cloud cover (t), temperature (t), transaction time (t), day of the week (t) as output values the price of electricity (t+1) [5],

- MISO-type ARX parametric input-output model, in which the volume in each hour of the day (t) is assumed as input and the volume-weighted average price ee (t + 1) is assumed as output [14].

The model used in this work is an input-output model of the MIMO type using ANN, in which 24 volumes of energy in each hour of the day (t) were assumed as input values, and 24 volume-weighted average energy prices in each hour of the day (t+1) were assumed as output values.

4 Analysis of works by other authors

Taking into account the analysis of the impact of various factors on the volume and the average volume-weighted price, they were adopted as synthetic measures taking into account factors such as: political, technical, social, economic, etc.

Table 1 presents the results of the literature review from the point of view of the following characteristics: authors, type of modeling, data, time horizon, evaluation measures, input and output quantities, type of object, and computational environment.

Based on the seven items of literature on the subject analyzed in detail, listed in Table 1, it follows, among others, that:

Table 1. Selected list of modeling characteristics for selected works. Source: own work [3, 9, 11, 14, 19].

authors	type of modeling	data period	Time horizon	Selected evaluation metrics	Input and output quantities	Object type	Computing environmen t
K. Halicka [5]	MLP input-output (MISO) model	1.01 –31.12.2004	1.01– 31.01.2005	m.in. MAPE	7:168-30-1:1	DAM TGE S.A.	STATISTICA
R. Weron, A. Misiorek [23]	12 time series methods – including: ARX model	California (1999-2000), NordPool (1998-1999, 2003-2004)	A week, a day	WMAE	1-1	California , Nord Pool	-
S. Voronin [22]	Time series: incl. SARIMA, GARCH	1999 –2010	among others year, seasons, week	m.in. AMAPE	m.in. 1-1	Nord Pool	MATLAB, R
Marlęga R., [14]	Input - output model: ARX with ARX correction MISO(ARX,	2002-2020	Selected periods of 2020: month, quarter, half-year, year	MAPE	MLP:24-1	DAM TGE S.A.	MATLAB
Lago J., Marcjasz G., Schutter B., Weron R. [12]	Time Series: Deep Artificial Neural Network (CNN)	01.01.2013 - 24.12.2018	27.12.2016–24.12.2018 (two years)	MAPE	CNN: 241-24	Pennsylvania–New Jersey– Maryland, EPEX BE,	Python toolbox.
Ejdys J., Halicka K., Godlewska J [3]	Time series: Multilayer perceptron (Multiplicative Holt-Winters model	1.01.2012– 30.04.2013	01.05.2013 (day)	MAPE	Holt-Winters model: 1-1MLP: 7-2-1	DAM TGE S.A.	STATISTICA
T. Popławski, M. Weżgowiec [17]	Time series (Creeping trend)	01.05.- 31.07.2015	01-07.08.2015 (weak)	MAPE	1-1	TGE, Nord Pool	GRETL

1. In five cases, time series were used in the research, and in three cases, input-output models Multi Input Single Output (MISO) were used,
2. Parametric models were used in six cases and neural models in three cases,
3. Single Input Single Output (SISO) -type models were used in three cases, MISO-type models in three cases, and MIMO-type models in one case,
4. In two cases, the STATISTICA package was used as the computing environment, in two cases MATLAB, in one case Python and in one case GRETL.

5 Behavior of the selected network

The research on modeling the DAM system included, among others considering the Radial ANN of the Generalized Regression Neural Networks type (Figure 1) due to its approximation properties. Its characteristic feature is that the output values of the first layer before entering the second layer are processed by the $normprod()$ function, which is the product of the output signal y^1 and the second layer weight matrix w^2 normalized by the sum of the elements of the y^1 vector, which in the case of individual signals can be express as follows:

$$\text{normprod}\left(y^{n}{}_{j}\right) = w_{ij}^{2} * \frac{y_j^1}{\sum y_j^1}, \tag{1}$$

where:
$y^{n}{}_{j}$ – normalized output from the hidden layer of neurons,
$w^{2}{}_{ij}$ – link between i-th neuron of output layer and j-th output from the hidden layer.

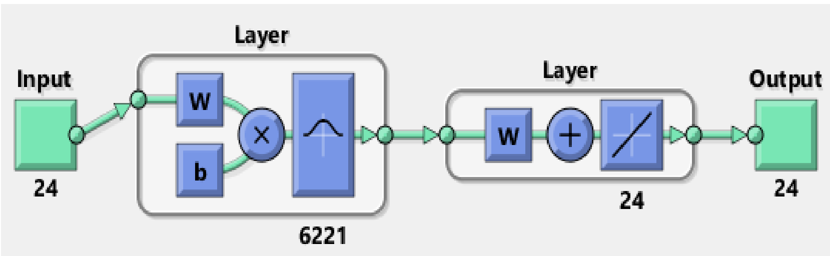

Figure 1. Architecture of the generalized Radial ANN used in the research experiment. Markings: Input (24) – input signals to the ANN, here: 24 signals representing the volume of electricity sold in individual hours of the day, Layer (6221) – hidden layer composed of 6221 neurons with a radial neuron activation function $newgrn()$, (each input signal is represented by another neuron), Layer (24) - the output layer of neurons composed of 24 output neurons with a linear neuron activation function $purelin()$, Output (24) - output signals from the ANN (here: 24 signals being the average price weighted by the volume of electricity sold every hour of the day. Source: Own work in the MATLAB and Simulink environment [2].

What is worth noting in this type of network is the ratio of the number of neurons in the hidden layer to the MSE error value. The analyzes show that it decreases as the attraction radius decreases. Figure 1 shows the architecture of a generalized Radial ANN with 6221 neurons in the hidden layer, which represent the number of all training pairs. The relationship between the MSE error value and the number of neurons in the hidden layer is shown in Figure 2.

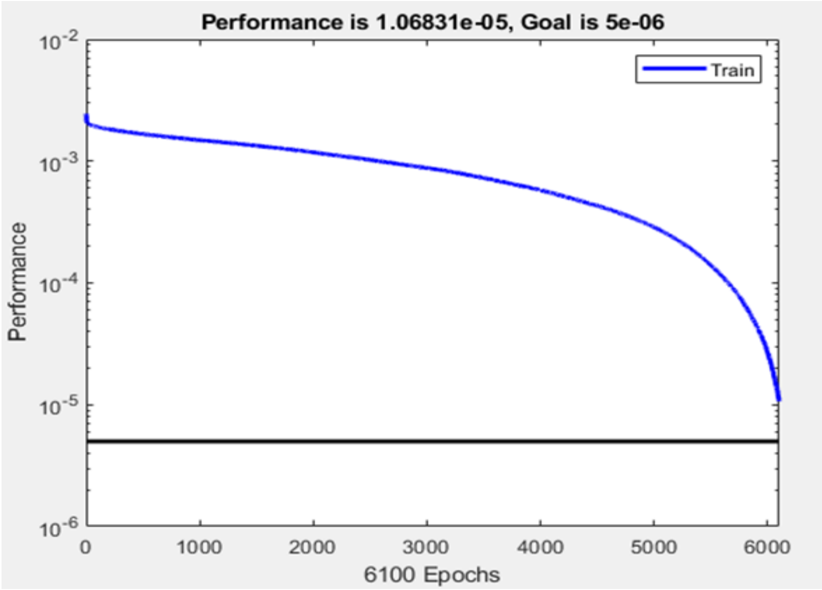

Figure 2. Dependence of the MSE error in relation to the number of neurons in the hidden layer in the Radial network. Markings: x-axis – number of neurons in the hidden layer, y-axis – MSE error value. Source: Own work in the MATLAB and Simulink environment [2].

Selected MSE error values depending on the number of neurons in the hidden layer are presented in Table 2.

Table 2. Selected MSE train error values depending on the number of neurons in the hidden layer. Source: own study in MATLAB and Simulink [2].

Number of neurons in the hidden layer	MSE error value	Number of neurons in the hidden layer	MSE error value
1	0.00244929	50	0.00196271
500	0.00166496	1000	0.00148653
2000	0.00118	3000	0.000878569
5000	0.000286215	6100	$1.06831*10^{-5}$

The analysis of the results obtained in Table 2 indicates that the MSE error decreases with the increase in the number of neurons in the hidden layer and reached the value of 1.06831*10-5 for 6100 neurons in the hidden layer. Taking into account the dependence of the number of neurons in the hidden layer on the MSE value, the maximum number of neurons, i.e. 6100, was adopted for further work

6 Testing the influence of the number of hidden layers

The impact of the number of hidden layers on the quality of the neural model based on another tested network that was taken into account in the modeling process of the DAM system, namely the Perceptron ANN, was also analyzed. Six networks were trained and the results of the decrease in training error in individual epochs, expressed as MSE, are presented in Figure 3.

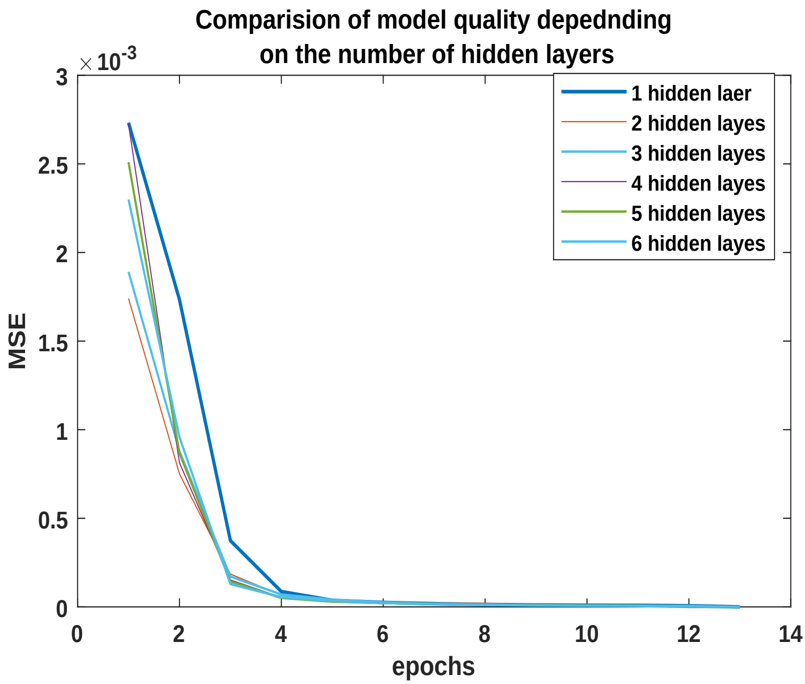

Figure 3. MSE error values depending on the number of hidden layers in the Perceptron ANN. Own development in the MATLAB environment [2].

As can be seen in Figure 3, the number of hidden layers does not have a major impact on the quality of the network measured by the MSE meter, regardless of the number of hidden layers in the training process, the trained model of the DAM system was achieved learning goal in epoch 4-6. Table 3 presents the MSE values in the fifth learning epoch for individual neural models with different numbers of hidden layers

The influence of the learning methods on the quality of the neural model based on Perceptron ANN was also analyzed. Eight learning models were analyzed, and used as models for learning weights and biases in initial propagation. The results of the research are presented in Table 4.

As can be seen in table 4 the most effective among the others comparing methods is the Levenberg-Marquardt method.

Table 3. MSE value for the 5th epoch of training the Perceptron ANN for a different number of hidden layers. Source: Own work in the MATLAB and Simulink environment using own m-files and DLT [2].

Number of hidden layers	The **value of MSE in the 4th epoch**
1 layer hidden	3.56045694324233e-05
2 hidden layers	3.61292021435581e-05
3 hidden layers	4.92057447000402e-05
4 hidden layers	3.86536498812968e-05
5 hidden layers	4.59909932288910e-05
6 hidden layers	5.16455410238103e-05

Table 4. MSE value for various learning methods Source: Own study in MATLAB and Simulink environment [2].

Learning method	**MSE value**
Levenberg-Marquardt	6.45200861044583e-06
BFGS Quasi-Newton	0.000347970045087015
Resilient Backpropagation	8.43831919196286e-05
Scaled Conjugate Gradient	6.61551823284186e-05
Conjugate Gradient with Powell/Beale Restarts	8.43881871135740e-05
Fletcher-Powell Conjugate Gradient	7.35838787923228e-05
Polak-Ribiere Conjugate Gradient	6.59734554615669e-05
One Step Secant	0.000305660835847778

7 A proposal to improve the model based on the implementation of the Evolutionary Algorithm

After training the ANN with the DAM system (neural model), an attempt was made to improve it by using the Evolutionary Algorithm (AE). In the theory of genetic and evolutionary algorithms, there are many different practical solutions in terms of their program implementation. Historically, from work on genetic algorithms [11], through evolutionary algorithms and evolutionary programming [9], to evolutionary strategies, including attempts to automate the writing of LISP programs (genetic programming) [10]. However, so far there is no uniform nomenclature, and attempts are being made to finally unify the above-mentioned names. methods under one name: evolutionary algorithms [1].

In the literature, evolutionary algorithms are often used to improve models of neural systems [4, 15, 20, 21]. Most often, in ANNs, the values of the weight and bias matrix require improvement. Individual AE methods are then used, including: the method of building the Initial Population, which in subsequent stages becomes the Parent Population, the crossbreeding method, the mutation method, the adaptation (robustness) method and the selection method, the selection of which for improving the ANN is based on their most explicit design and minimizing CPU load (the same methods were used in the quantum-inspired AE).

Designing the Initial Population (PP)

The design of the Initial Population was based on a Perceptron ANN trained with the neural model of the DAM TGE S.A. system. based on data from transactions that occurred in the period 01/01/2019 to 30/06/2019 (i.e. 181 days) for all hours of the day.

Improving the neural model of the DAM system

The improvement of the neural model of the DAM system is based on improving the accuracy of the weight values of both weights matrices of the Perceptron Artificial Neural Network and on improving the accuracy of their mutual relations between the neuron layers, and thus, in a sense, on improving the mutual matching of the values of the elements of both weight matrices. The accumulated knowledge contained in the weight matrices is tuned to the DAM system using the Evolutionary Algorithm.

Construction of the genotype structure

In this case, it is assumed that the individuals of the population are composed of a single chromosome, in which subchromosomes represented by both weight matrices $\mathbf{W}^1$ and $\mathbf{W}^2$ are immersed [1], with the weight values ($w_{ij}{}^n$) of individual rows of both matrices immersed in them, which can be formally written as follows:

$$\mathbf{Ch}_i(t) = [\mathbf{W}^1\mathbf{W}^2], \tag{2}$$

where:

$\mathbf{W}^1$=[[$w^1{}_{1,1}$ $w^1{}_{1,2}$ ... $w^1{}_{1,24}$] [$w^1{}_{2,1}$ $w^1{}_{2,2}$ ... $w^1{}_{2,24}$] ... [$w^1{}_{24,1}$ $w^1{}_{24,2}$ ... $w^1{}_{24,24}$]],
$\mathbf{W}^2$= [[$w^2{}_{1,1}$ $w^2{}_{1,2}$ ... $w^2{}_{1,24}$] [$w^2{}_{2,1}$ $w^2{}_{2,2}$ ... $w^2{}_{2,24}$] ... [$w^2{}_{24,1}$ $w^2{}_{24,2}$ ... $w^2{}_{24,24}$]],
$\mathbf{Ch}_i$(t) – i-th chromosome with values at time t.

Using the definition of chromosome structure (2), it is possible to generate a PP composed of individuals with the structure of a single chromosome containing 1,152[2] genes, which in the considered case are the values of individual chromosome elements ($w_{ij}{}^c$). The values of subsequent genes for all chromosomes are generated based on values drawn from the intervals created as follows:

$$w_{ij}{}^c = w_{ij}{}^n \pm \Delta w_{ij}{}^n, \tag{3}$$

where:

$\Delta w_{ij}{}^n$ – increase in the weight value as a result of increased accuracy of the weight value (e.g. by one degree of precision) in relation to the weight value).

The generated chromosome structure was used in the Initial Population to create 10,000 individuals, which, as a result of AE operation, became the Parent Population using the AE algorithm.

---

[1] The concept of immersion was introduced, among others, in [24], which is understood as the minimum number of dimensions (geometric coordinates) necessary to reflect (describe) the system. The essence of the problem is to determine such a space where the so-called phenomenon does not occur. false neighbors, i.e. points that are close to each other in n-dimensional space may be located far from each other in n+1-dimensional space. The minimum number of dimensions when this phenomenon does not occur is called the minimum immersed dimension

[2] two matrices x matrix dimension: 2 x (24 x 24) = 2 x 576.

7.1 Crossbreeding method and mutation method

Evolutionary algorithms use various crossover and mutation methods depending on the type of research experiment being conducted. The most frequently used crossbreeding methods to improve weight values in AE include single-point and multi-point crossbreeding of individuals, and mutation methods include changing the sign of the weight value, increasing/decreasing the weight value by an order of magnitude, etc.

The crossbreeding method (single or multi-point) is used with a probability of 0.1 ≤ pk ≤ 1.0, and its operation consists in cutting two individuals from the population (connected in pairs) at a randomly selected crossbreeding site (locus[3]) and then replacing the cut off parts of both individuals between them. For the purposes of this work, the single-point crossover method, most frequently used to improve the value of weights in ANNs, was selected.

The mutation method is usually used with a low probability of 0.01 ≤ pm ≤ 0.09 (usually an order of magnitude lower than the crossover probability). The result of the method is an increase in the diversity of the populations that are subject to assessment and succession. For the purposes of this work, the mutation method was chosen, which involves changing the value of the randomly selected weight by an order of magnitude.

7.2 Adaptation (robustness) method

The adaptation method in AE is used to assess the degree of adaptation of individuals in each cycle of the algorithm. In the literature on the subject, there are several concepts for building functions used in the method of adapting individuals to the structure of population individuals, but there are no sufficiently developed methods for selecting adaptation functions.

For the purposes of this research, the discrepancy function resulting from the degree of mismatch between the neural model and the DAM system was assumed as the fitness function (robustness function). The measure of an individual's adaptation to the environment is the difference between the expected output values from the neural model and the average output values from the DAM system, determined as the Mean Square Error (MSE) calculated according to equation (4).

$$MSE = \frac{1}{n} \sum_{i=1}^{n} \left(y_i^s - y_i^n\right)^2, \tag{4}$$

where:

y_i^s - exit from the DAM system (volume-weighted average price ee),

y_i^n – output from the neural model of the DAM system (volume-weighted average price ee).

7.3 Selection method

Various selection methods are used in AE, such as: roulette method, tournament method, ranking method, elite method and many other methods [1].

[3]Locus – a position indicating the location of a given gene on the chromosome.

The roulette method (also called the proportional method) involves determining for each individual the probability of its reproduction, determined as the ratio of its fitness value to the sum of the fitness values of all individuals. Individuals occupying a "larger" probability area have a greater chance of being selected into the next population.

The ranking method involves assigning an appropriate weight (rank) to each individual, which characterizes its adaptation value. The population marked in this way is sorted according to the assigned values (usually expressed as a set of natural numbers). After assigning ranks, a random variable is determined that assigns the probability of selecting a given individual from the population depending on the previously obtained rank.

The tournament method involves selecting a specific number of individuals from the available population (each individual is equally likely to be drawn), and then the individual with the highest fitness value is selected from the drawn group. This process is repeated until the entire offspring population is selected.

The elite method consists in sorting by the fitness value of the Parent Population, then deterministically selecting a certain part of the best-adapted parental individuals to the new population and selecting the remaining population using the selected method. For the purpose of improving the DAM system model, three AE selection methods were examined:

1. Elite,
2. Roulettes,
3. Tournament.

The obtained MSE error curves for the subsequent selection methods are presented in Figures 4 to 6, respectively.

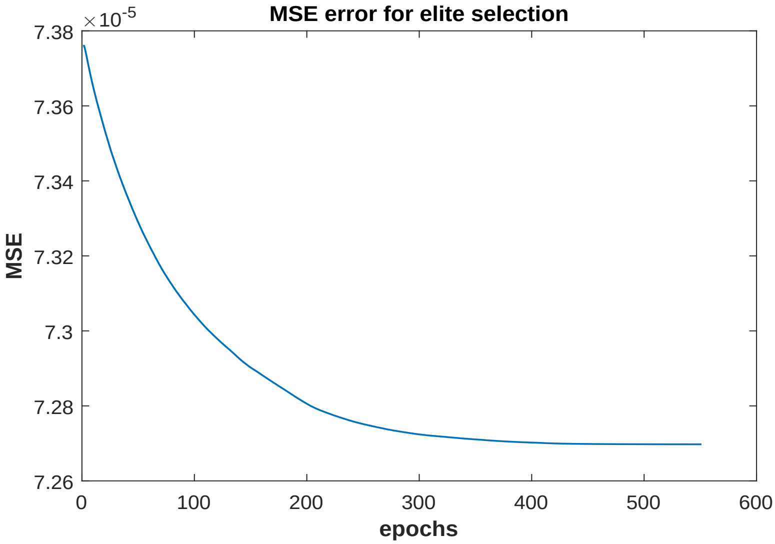

Figure 4. The course of the MSE error when the elite selection method is used in AE. Source: Own study in the MATLAB environment [2].

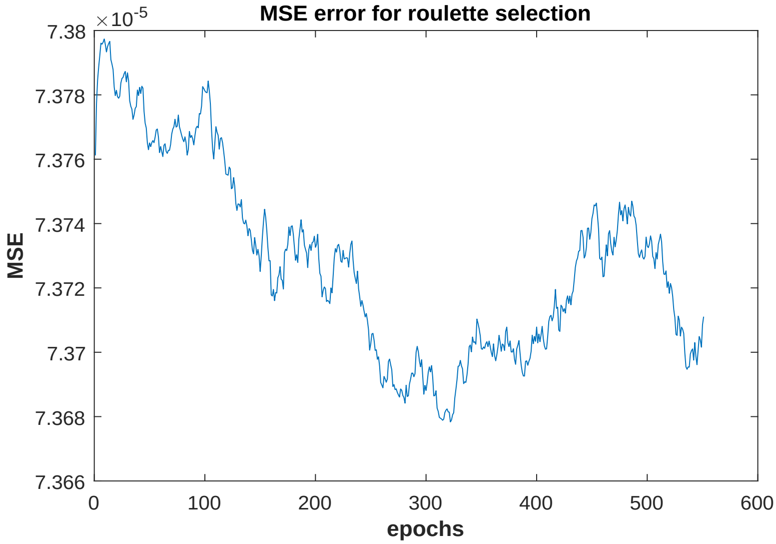

Figure 5. The course of the MSE error when using the roulette method in AE. Source: Own study in the MATLAB environment [2].

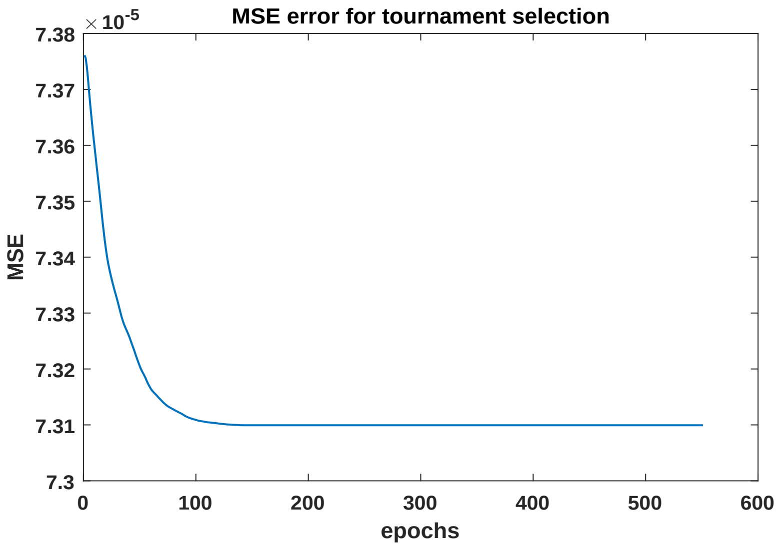

Figure 6. MSE error progression when using the tournament selection method in AE. Source: Own study in the MATLAB environment [2].

The analysis of the obtained MSE error patterns shows that the best course is characterized by the AE algorithm with tournament selection, therefore, for the purposes of the research carried out in this work, the tournament method was adopted as the selection method without returning the randomly selected individuals, due to the high selective pressure[4].

The relationship between ANN and AE in the process of improving ANN weights is shown in Figure 7.

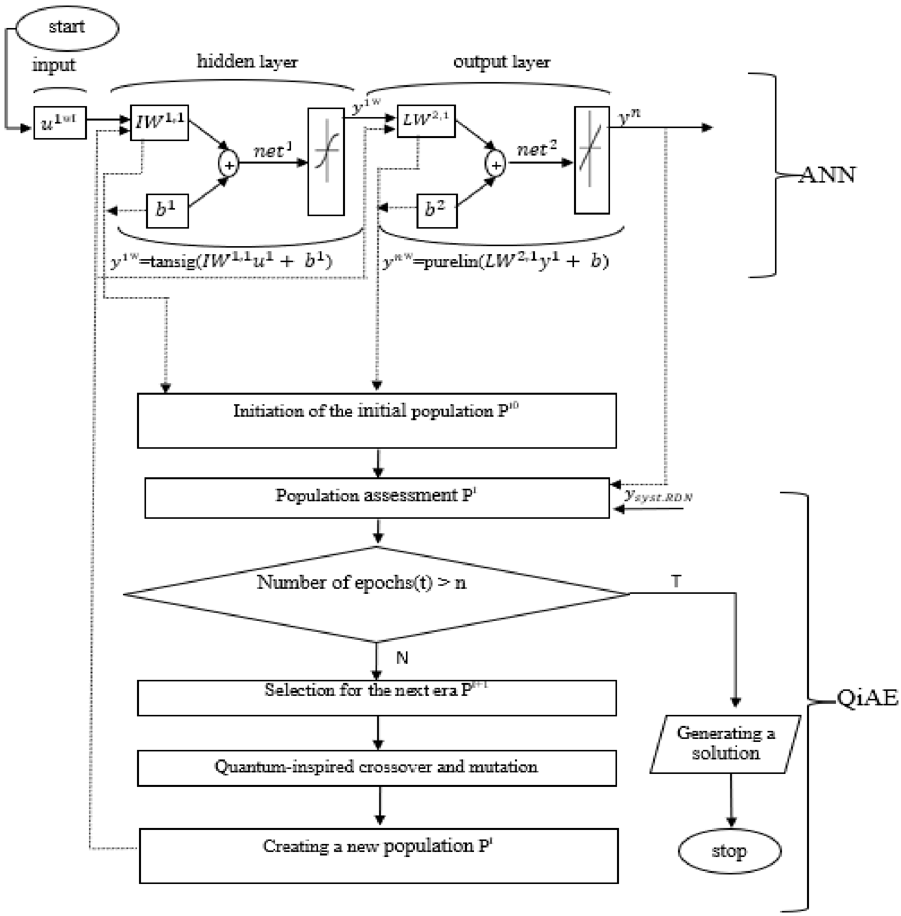

Figure 7. A scheme for using the Evolutionary Algorithm to improve the weights of the Perceptron Artificial Neural Network. Designations: dotted line – relation between MATLAB and AE, continuous line course within both models. Source: Own work in MATLAB and Simulink notation [2].

[4]Selective pressure - this is the tendency of an AE to improve the average fitness of a population during the next evolutionary cycle.

After selecting 10,000 winning chromosomes, they were determined to adapt to the DAM system. To improve the neural model, AE runs for 50 generations. The results of the improvement of the neural model of the DAM system are presented in Figure 8.

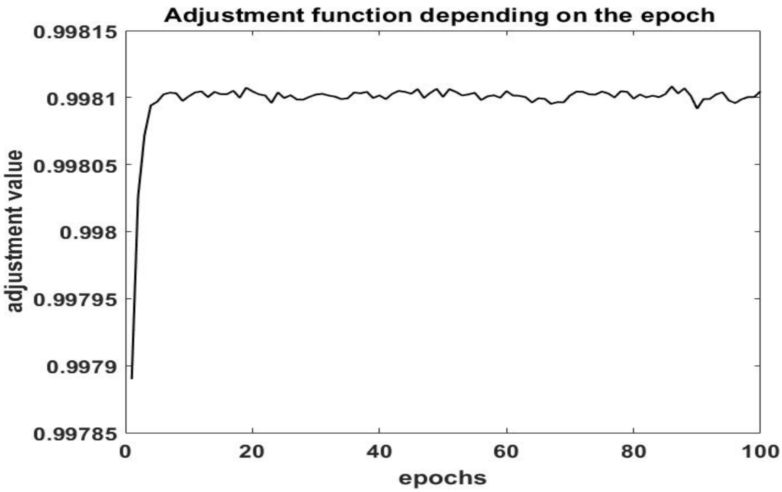

Figure 8. The course of the adjustment function depending on the generation (epoch) in the Evolutionary Algorithm. Designations: epoch – the number of the next population obtained as a result of crossing, adjustment value – the value of adaptation of the neural model to the DAM system. Source: Own study in the MATLAB and Simulink environment [2].

The possibility of improving the quality of ANNs by using AEs without evolutionary operators (crossing and mutation) was also analyzed, the results of this experiment are presented in Figure 9.

As can be seen in Figure 9, it turned out that no improvement in the quality of the ANNs was achieved in the case of multiple sampling. Studies were conducted for multiple sampling in numbers ranging from 1000 to 10,000 at a step of 100.

For comparison, the decrease in MSE error for SSN with improvement with AE using genetic operators is shown in Figure 10.

Checking the degree of improvement of the neural model as a result of the use of the Evolutionary Algorithm in relation to the DAM system model in the form of an Artificial Neural Network shows that the MSE error, which for the Perceptron ANN was 0.0242, for the ANN corrected with AE 0.0124, which indicates that it indicates that the SNN corrected with AE obtains results closer to the real values. You can see that the MSE error of the evolutionarily corrected ANNs has almost doubled .

8 Conclusions

The work focused on improving the price model on the Day-Ahead Market (DAM) of the Electricity Exchange (TGE S.A.) using neural networks (ANN). Several key aspects of modeling were shown, such as the architecture of neural networks, types of networks, and the use

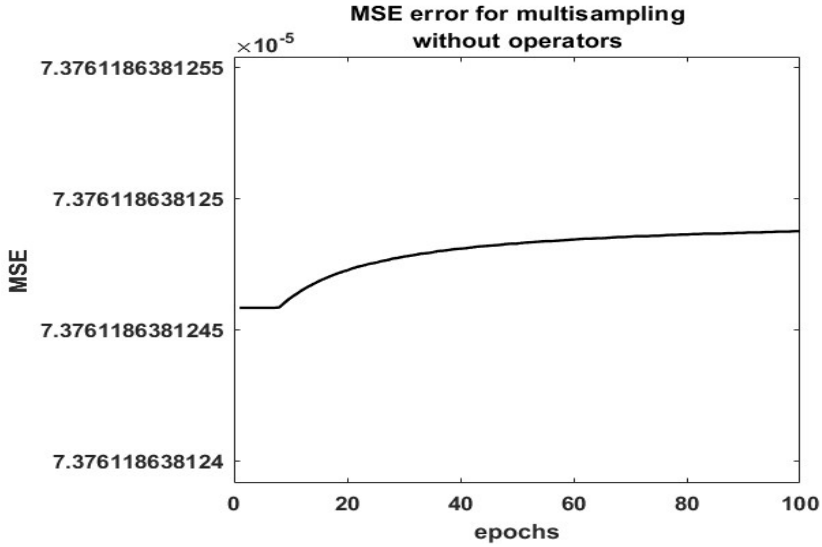

Figure 9. The Multiple sampling in AE (without the use of genetic operators). Source: Own study in the MATLAB and Simulink environment [2].

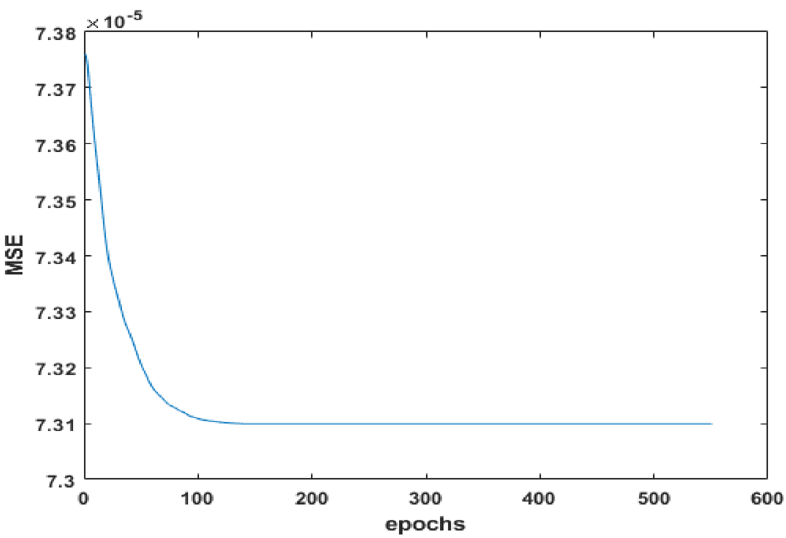

Figure 10. AE with the use of genetic operators. Source: Own study in the MATLAB and Simulink environment [2].

of Evolutionary Algorithms to optimize models. Unlike most works of this type operating on time series, a MIMO (Multiple Input Multiple Output) model based on ANN was proposed. The input data include electricity volumes during individual hours of the day, and the output data are weighted average energy prices for the next day. A literature review was conducted on various methods of modeling electricity prices, such as time series and input-output models (MISO, MIMO). Computing environments and model evaluation methods were also compared. Details of the Radial ANN architecture is presented, illustrating the dependence of the MSE error on the number of neurons in the hidden layer. The influence of the number of hidden layers on the quality of the model was also discussed. It was proposed to improve the model using Evolutionary Algorithms: The possibility of optimizing the ANN model using Evolutionary Algorithms was investigated. Experiments were carried out with various training methods in the context of MSE error reduction. The paper demonstrates the usefulness of improving the DAM system model based on the proposed ASN-AE hybrid model.

References

1. Arabas, J. (2016), Wykłady z Algorytmów Ewolucyjnych (Lectures on Evolutionary Algorithms). WNT, Warszawa, p. 303.
2. Beale, M. H. [at al.] (1992-2019), Neural Network Toolbox™ User's Guide, The MathWorks, Inc. p. 846.
3. Ejdys, J., Halicka, K., Godlewska, J., (2015), Prognozowanie cen energii elektrycznej na giełdzie energii (Forecasting electricity prices on the energy exchange), Wydawnictwo Politechniki Łódzkiej, Zeszyty Naukowe. Organizacja i Zarządzanie, zeszyt 77, pp. 53–61.
4. Bulatovic, L. B., Sukovic, G. R. (2018), On applying evolutionary algorithms for hybrid neural networks' architecture synthesis, 23rd International Scientific-Professional Conference on Information Technology, Zabljak, pp. 1-4.
5. Halicka, K.,(2006), Skuteczność prognozowania w zarządzaniu transakcjami na giełdzie energii elektrycznej (The effectiveness of forecasting in managing transactions on the electricity exchange). Rozprawa doktorska pod kierunkiem prof. dr hab. inż. J. Nazarko, UW, Warszawa p. 207.
6. Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9, pp. 1735-1780.
7. Holland, J. H. (1975). Adaptation in Natural and Artificial Systems. University of Michigan Press p. 194.
8. Hopfield, J.J., (1982). Neural networks and physical systems with emergent collective computational abilities. Proc. Natl Acad. Sci. USA. pp. 2554–2558. doi: 10.1073/pnas.79.8.2554
9. Koza, J. R. (1992). Genetic Programming: On the Programming of Computers by Means of Natural Selection p. 840.
10. McCarthy, J. (1960). Recursive Functions of Symbolic Expressions and Their Computation by Machine, Part I. Communications of the ACM, vol. 3, issue 4 pp. 184-195.
11. Moody, J., & Darken, C. (1989). Fast Learning in Networks of Locally-Tuned Processing Units. In D. S. Touretzky (Ed.), Advances in Neural Information Processing Systems 1, pp. 281-288.
12. Lago, J., Marcjasz, G. [et al.], (2021) Forecasting Day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark. Aplayed Energy Vol. 293, p. 21.
13. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-Based Learning Applied to Document Recognition. Proceedings of the IEEE, 86, pp. 2278-2324.

14. Marlęga, R. (2021) Comparative Study of the Identification Methods of the Management System of the Day-Ahead Market of Polish Energy Market S.A. Studia Informatica. System and Information Technology, Nr 1-2 pp. 67-86.
15. Michalewicz, Z. (2011) Genetic Algorithms + Data Structures = Evolution Programs, Springer-Verlag, p. 387.
16. Mielczarski, W. (2000) Rynki energii elektrycznej. Wybrane aspekty techniczne i ekonomiczne (Electricity markets. Selected technical and economic aspects), ARE, Warszawa, p. 308.
17. Popławski, T., Weżgowiec, M., (2017), Prognozowanie Cen Energii Na Rynku Spot Giełdy Nord Pool i TGE (Forecasting Energy Prices on the Spot Market of the Nord Pool and TGE Stock Exchanges), Rynek Energii 1/2017, p. 11.
18. Powell, M. J. D. (1987). Radial Basis Functions for Multivariable Interpolation: A Review. In J. C. Mason & M. G. Cox (Eds.), Algorithms for Approximation, pp. 143-167.
19. Rosenblatt, F. (1958) The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review, 65, pp. 386–408
20. Ruciński, D. (2016) Neural-evolutionary Modeling of Polish Electricity Power Exchange. Electrical Power Networks, EPNet 2016, Katedra Energoelektryki PWr, Institute of Power Engineering and Control Systems of Lviv Polytechnic National University, Ukraine and O/Wrocławski SEP, XPlore Digital Library, Szklarska Poręba 2016 r., pp. 1-6.
21. Tchórzewski, J. (2021) Metody sztucznej inteligencji i informatyki kwantowej w ujęciu teorii sterowania i systemów (Methods of artificial intelligence and quantum computing in terms of control and systems theory), Wydawnictwo Naukowe Uniwersytetu Przyrodniczo-Humanistycznego, p. 343.
22. Voronin, S. (2013) Price Spike Forecasting in a competitive day-ahead energy market, Acta Universitatis, Lappeenrantaensis 530, 2013, p. 181.
23. Weron, R., Misiorek A.(2008) Forecasting spot electricity prices: A comparison of parametric and semiparametric time series models, MPRA Paper, No.10428, Monachium, p. 23.
24. Zieliński, J. S. [et al.] (2000) Inteligentne systemy w zarządzaniu, Teoria i praktyka (Intelligent systems in management, theory and practice). WN PWN, Warszawa, p. 349.
25. Zhang, H., Hu, W. [et al.] (2024) A Temporal Convolutional Network Based Hybrid Model for Short-term Electricity Price Forecasting,CSEE Journal Of Power And Energy Systems, Vol. 10, No. 3, pp.1119-1130.