HYBRID APPROACHES TO MAXIMUM POWER POINT TRACKING (MPPT) BASED ON ARTIFICIAL INTELLIGENCE: SYNERGY OF METHODS TO INCREASE EFFICIENCY

Javad Vaqif Najafli
Azerbaijan Technical University, Baku, Azerbaijan: cavad.necefli@aztu.edu.az
https://orcid.org/0009-0008-3396-5657

Abstract. This study investigates the performance of hybrid Maximum Power Point Tracking (MPPT) algorithms for photovoltaic systems using neural networks (NN), genetic algorithms (GA), and particle swarm optimization (PSO). Three hybrid models—NN-GA, PSO-GA, and NN-PSO—were evaluated based on response time, maximum power, and robustness to noise. Results revealed distinct trade-offs among the models. The NN-GA model exhibited the shortest response time of 0.0036 s/prediction, but its accuracy significantly degraded with increasing noise levels. The PSO-GA model demonstrated the highest maximum power output of 100.24 W and superior robustness to noise, with a mean absolute error (MAE) of 0.2062 at 30% noise. The NN-PSO model provided balanced performance, achieving a maximum power of 45.90 W and demonstrating the lowest MAE of 0.1537 at 20% noise. These findings highlight the transformative potential of hybrid MPPT methods in enhancing both the adaptability and performance of solar energy systems across diverse and challenging operational conditions.
Keywords: *Maximum Power Point Tracking, neural networks, genetic algorithms, particle swarm optimization, hybrid models.*

Introduction

Solar energy plays an important role in the development of renewable energy sources due to its environmental friendliness, availability, and cost-effectiveness. However, the low conversion efficiency of photovoltaic (PV) systems, ranging from 9 to 17%, requires the use of power optimization technologies to improve their efficiency. Maximum Power Point Tracking technology provides an opportunity to maximize the output power of solar panels by adapting them to changing climatic conditions [1, 2, 3].

Traditional methods such as perturbation and observation (P&O) and incremental conductivity (INC) algorithms are widely used due to their simplicity and low implementation cost. However, these methods have limitations such as steady-state oscillations and efficiency degradation under rapidly changing solar radiation [4, 5]. In recent years, much attention has been paid to the implementation of intelligent methods, including neural networks, logic controllers, and particle optimization algorithms. These approaches demonstrate improved tracking performance, robustness to data noise, and adaptation to complex operating conditions [6, 3].

Several studies have been conducted to compare MPPT methods to identify their strengths and weaknesses. For example, Sarvi and Azadian's study classifies algorithms into traditional, smart, and hybrid, noting that hybrid approaches have an advantage in non-uniform shading conditions [3]. Eltawil and Zhao's work provides a detailed discussion of the advantages and disadvantages of smart methods over traditional ones, highlighting their potential to improve the efficiency of solar systems [2].

This study aims to investigate and evaluate state-of-the-art MPPT algorithms applied to photovoltaic systems, focusing on comparing their performance characteristics such as response time, robustness to data noise, and overall energy extraction efficiency, thereby identifying promising areas for further development.

Materials and Methods

Maximum Power Point Tracking technology is designed to maximize the energy extracted from solar panels despite changes in environmental conditions (e.g. sunlight intensity, temperature, shading). Each solar panel has a Maximum Power Point (MPP) — the combination of voltage and current at which the greatest power is produced. However, due to the nonlinear characteristics of solar cells and changing conditions, this point changes throughout the day.

MPPT algorithms measure the panel voltage (U) and current (I) to calculate the output power $P = U \cdot I$ P=U·I. MPPT analyzes the current power and compares it to previous measurements. If the power increases, the load settings change in the same direction. If it decreases, the direction is adjusted. MPPT controllers continuously adjust the panel voltage and current to achieve the optimal power point, even as weather conditions change. The implementation of MPPT provides maximum energy efficiency of solar systems, increasing the output power by 10-30% compared to systems without MPPT [7].

This study aims to evaluate the capabilities of artificial intelligence (AI) to improve the efficiency of MPPT. It includes several stages: first, the classical Perturb and Observe (P&O) algorithm is simulated, which is the basis for performance analysis. The simulator generates random voltage and current values in the range of 0-60, calculates the power and records the results. Performance is assessed by average, maximum and minimum power. The indicators of such AI models as neural networks, decision trees, genetic algorithms, particle swarm optimization (PSO) are compared.

The following criteria are selected for comparing the models:

- Response Speed: The time it takes to adapt to changing conditions.
- Power Extraction Efficiency: The maximum power that a model can achieve.
- Robustness to Data Noise: The ability to maintain performance in the presence of variations or errors in the input data.

Neural Network. A neural network is a composite mathematical model that transforms input data into output through a system of interconnected nodes (neurons) organized into layers [8]. For MPPT problems, neural networks are trained on data including voltage, current, and power to predict optimal voltage and current values. The general formula for a single neuron is: The output y_i of a neuron in layer l is defined as:

$$y_i^{(l)} = f(\sum_{j=1}^{n} w_{ij}^{(l)} x_j^{(l-1)} + b_i^{(l)}), \quad (1)$$

where $x_j^{(l-1)}$ is the output of the j-th neuron from the previous layer, $w_{ij}^{(l)}$ is the weight of the connection between neurons, b_i^l is the bias of the current neuron, $f(\cdot)$ is the activation function (in our case, it is ReLU), n is the number of inputs for the current neuron. The network optimizes the parameters (weights w_{ij} and biases b_i) based on the loss function L, which measures the discrepancy between the predicted output $\tilde{y}$ and the true value y. An example of a loss function:

$$L = \frac{1}{N}\sum_{i=1}^{N}(y_i - \tilde{y}_i)^2, \quad (2)$$

where N is the number of training examples.

Gradient descent is used to minimize the loss function:

$$w_{ij}^{(l)} \leftarrow w_{ij}^{(l)} - \eta \frac{\delta L}{\delta w_{ij}^{(l)}}, \quad (3)$$

$$b_{ij}^{(l)} \leftarrow b_{ij}^{(l)} - \eta \frac{\delta L}{\delta b_{ij}^{(l)}}, \quad (4)$$

where η is the learning rate. For the voltage and current prediction task, a linear activation function is used in the output layer to ensure continuous output. The network is tuned based on synthetic data representing the voltage, current, and power of solar panels under different conditions.

Genetic algorithms. These are heuristic optimization methods that imitate the processes of natural selection described by the theory of evolution. These algorithms are suitable for finding optimal or approximately optimal solutions to complex problems. The genetic algorithm is based on the use of a population of possible solutions to the problem, represented as chromosomes that encode the solution parameters [9]. The quality of each solution is assessed using the fitness function, which determines the efficiency of the corresponding solution. During the operation of the algorithm, selection is carried out, in which chromosomes with the highest values of the fitness function are chosen to participate in reproduction. The creation of a new generation includes a combination of parent chromosomes using the crossover operation, as well as the use of mutation, which introduces random changes to maintain the genetic diversity of the population. After this, the new generation replaces the previous one. The algorithm begins with initialization, during which an initial population of random solutions is created. At each iteration step, the fitness function is assessed, parents are selected, and crossover and mutation are performed to generate a new generation. The process continues until a specified number of iterations is achieved or the target optimization criterion is met.

A chromosome is a string of parameters $x = (x_1, x_2, \dots, x_n)$, where x_i is the parameter value. For MPPT, these are voltage and current. The fitness function evaluates the quality of each solution. In MPPT, the fitness function is defined as: $f(x) = P(x)$, where P is the output power of the solar panel, depending on the parameters x. Then selection occurs – the probability of choosing the i-th chromosome p_i is proportional to its fitness:

$$p_i = \frac{f(x_i)}{\sum_{j=1}^{N} f(x_j)},$$

where N is the population size. The crossover generates offspring o_1, o_2 by combining parent chromosomes p_1, p_2. An example of a single-point crossover:

$$o_1 = (p_1[1{:}\,k], p_2[k+1{:}\,n]), \tag{5}$$
$$o_2 = (p_2[1{:}\,k], p_1[k+1{:}\,n]), \tag{6}$$

where k is the split point.

Mutation introduces a random change in chromosome x: $x_i' = x_i + \delta$, where δ is a random value from a given range. A new population is formed from the offspring, or combined with a part of the old population to preserve elite solutions. The algorithm terminates if the specified number of iterations is reached or the fitness function value stops improving.

Particle Swarm Optimization (PSO).

It is an optimization method inspired by the social behavior of flocks of birds or schools of fish. The algorithm uses particles that move through a solution space to find the optimal value of an objective function [10]. Each particle represents a potential solution to the problem and moves through the solution space based on its past experience and social interactions. To operate, the PSO algorithm first defines an initial population of particles, each with a position x_i and a velocity v_i chosen at random. Each particle is evaluated using an objective function $f(x)$ to determine its "quality". The velocity and position are updated according to the formula: $v_i = \omega v_i + c_1 r_1 (p_i - x_i) + c_2 r_2 (g - x_i)$, where ω is the inertia coefficient controlling the influence of the previous velocity, c_1, c_2 are the learning coefficients for the personal and social components, r_1, r_2 are random numbers in the range [0,1] providing stochasticity, p_i is the best position of the particle, g is the best position among all particles in the population. The position of the particles is updated according to the formula $x_i = x_i + v_i$. The algorithm repeats the process of updating the velocity and position until the stopping criterion is reached.

In the context of MPPT, PSO is used to find the optimal combination of voltage and current that provide the maximum power of the solar panel. Each particle represents a possible combination of (V, I), and the objective function is the output power: $f(V, I) = P = V \cdot I$

Experiments

In the course of the study, three hybrid MPPT models were developed and tested: NN-GA, PSO-GA and NN-PSO. Their performance was evaluated by key indicators: maximum power, response time and noise immunity, the results of which are presented in Table 1 and the corresponding figures. First of all, synthetic data for solar panels with different noise levels (10%, 20%, 30%) were created, successfully generated and saved in a csv file (Figure 1). This includes voltage, current and power data, which will be used to test the models.

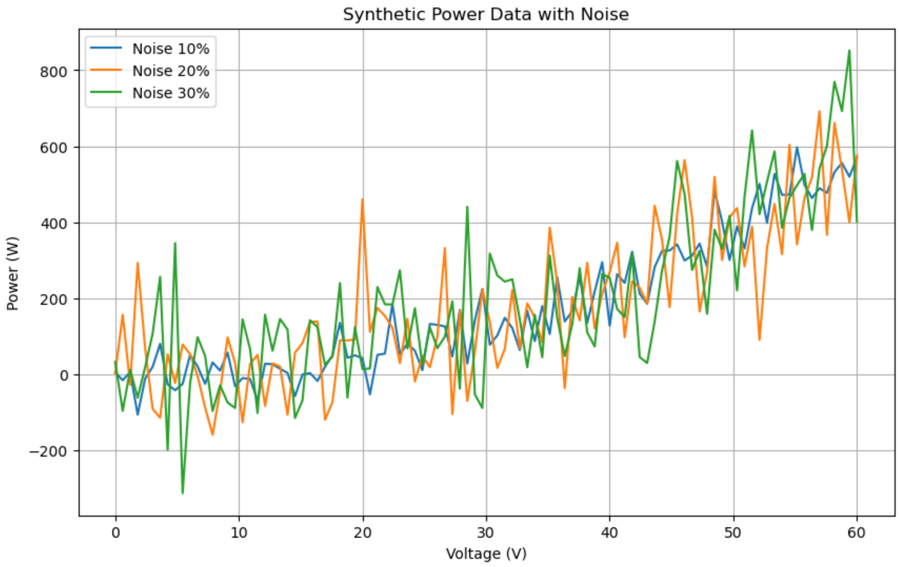

Figure 1. Power Data with Noise

NN-GA. Next, a hybrid NN-GA model was implemented using this data. The neural network architecture included an input layer with 2 neurons corresponding to the input variables, 2 hidden layers with 10 neurons each and the ReLU activation function, and an output layer with 1 neuron and a linear activation function. This configuration was chosen to efficiently model nonlinear dependencies between input and output data. A genetic algorithm was used to optimize the network parameters. An individual in the algorithm was a one-dimensional array containing all the weights and biases of the network. The population size was 20, and the number of generations was 5. The crossover operator was implemented as a mixture with a coefficient of 0.5 for the efficiency of combining parameters between individuals. To prevent getting stuck in local minima, a Gaussian mutation was used with the following parameters: mean 0, standard deviation 0.1, and the probability of mutation of each parameter 0.2. Tournament selection with tournament size 3 was used to select the strongest individuals for subsequent generations.

The settings were chosen to provide a balance between computational efficiency and model accuracy. Limiting the number of generations and population size allowed to speed up the optimization process while maintaining sufficient accuracy. The use of tournament selection and mutation facilitated exploration of the solution space and avoided getting stuck at suboptimal points.

A combination of a neural network and a genetic algorithm was chosen to provide a more efficient search for optimal model parameters. The genetic algorithm is capable of finding global optima in complex spaces, which complements traditional optimization methods such as gradient descent. The approach was chosen given the limited amount of data and the presence of noise.

When testing the model on data with a noise level of 20%, the following results were obtained: Test Loss 0.1328, MAE 0.3310

A value of 0.1328 Test Loss indicates a relatively low error, especially if the target values (power) were normalized in the range [0, 1]. In this case, the result can be considered good. A value of 0.3310 MAE is also moderate, but the accuracy can be improved. If the power is normalized, the absolute error of 33% is acceptable, but can be reduced by further optimization of the model.

PSO-GA. Next, a hybrid model of neural network parameter optimization was implemented based on a combination of particle swarm optimization (PSO) and genetic algorithm (GA). This model uses PSO to roughly tune the parameters, and GA performs their subsequent refinement, which allows for an effective combination of global search and local optimization. The neural network model included 2 hidden layers with 10 neurons each and the ReLU activation function. The following settings were used to optimize the PSO parameters: the inertia coefficient was 0.5, the cognitive and social component coefficients were 1.5, the particle population size was set to 20, and the total number of iterations was 10. The particles updated their positions based on their own and global success history to quickly explore the parameter space.

Based on the found solution, the particle swarm method passed the optimized parameters to the genetic algorithm, which refined them using crossover and mutation operations. The following parameters were chosen for the genetic algorithm: crossover probability of 0.5, mutation probability of 0.2, and the total number of generations was 5. The population size for the genetic algorithm was also 20.

These settings were chosen to provide a balance between computational efficiency and model accuracy, as well as to minimize the mean square error during the testing phase. Test results: Test Loss: 0.0484, Test MAE: 0.1714

NN-PSO. The last to be implemented was a hybrid neural network model, the parameters of which were optimized using the particle swarm optimization (PSO) algorithm. The architecture of the hybrid models is the same as in the pairs of the previous two.

Results: Test Loss 0.0411, MAE 0.1565

Final test and discussion

After training the models, their speed of adaptation to changes (response time), energy extraction efficiency (maximum achieved power) and noise immunity were tested.

The NN-GA model showed the maximum power of 35.82 W, which is inferior to other models, but compensated for this with the shortest response time - 0.0036 s / prediction. However, its accuracy significantly decreased with increasing noise level. For example, at a noise level of 30%, the loss was 0.3936, and the MAE reached 0.5910, as can be seen from the data in Table 1.

In contrast, PSO-GA demonstrated the highest power, reaching 100.24 W, and high noise immunity. At 30% noise level, its MAE was only 0.2062, as confirmed by Figure 2. However, the response time of this model was slightly higher than that of NN-GA, amounting to 0.0043 s/prediction.

The most balanced results were shown by the NN-PSO model. With a maximum power of 45.90 W and a response time of 0.0045 s/prediction, it demonstrated the best noise immunity among all models. For example, at 20% noise level, its MAE was only 0.1537, which is the lowest value among all the studied approaches.

The comparative analysis presented in Figures 3 and 4 confirms that the models using PSO cope with the power extraction task more efficiently, especially in high noise conditions. However, NN-GA remains preferable when minimizing response time is required (Figure 5).

Table 1

Performance of Models under Noise (MSE and MAE)

Model	Noise level (MSE)			Noise level (MAE)		
	0.1	0.2	0.3	0.1	0.2	0.3
NN-GA	0.4332	0.4371	0.3936	0.6193	0.6333	0.5910
PSO-GA	0.0494	0.0490	0.0690	0.1731	0.1727	0.2062
NN-PSO	0.0409	0.0388	0.0432	0.1565	0.1537	0.1611

Table 2

Models' MP and response time

Model	Maximum Power (W)	Response time (s/pred.)
NN-GA	35.82	0.0036
PSO-GA	100.24	0.0043
NN-PSO	45.90	0.0045

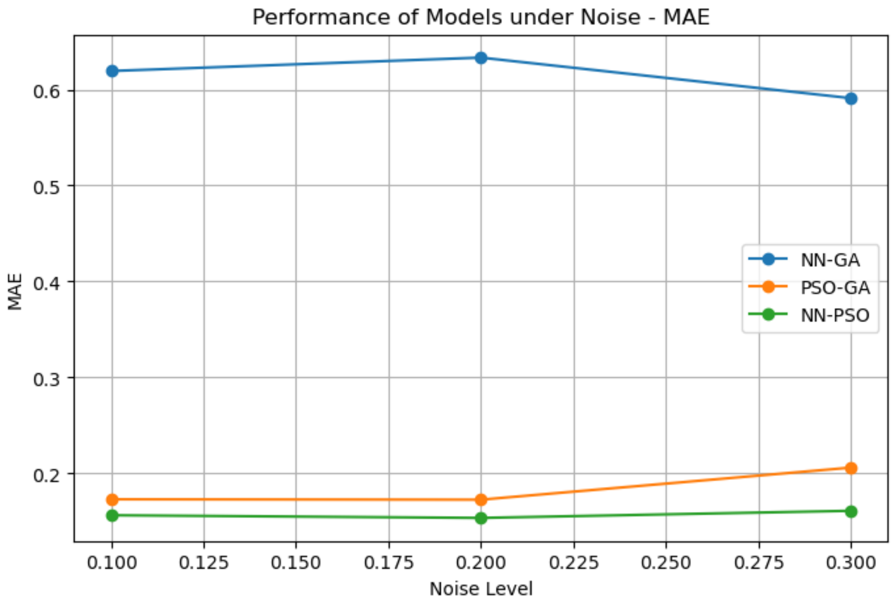

Figure 2. Performance of Models under Noise – MAE

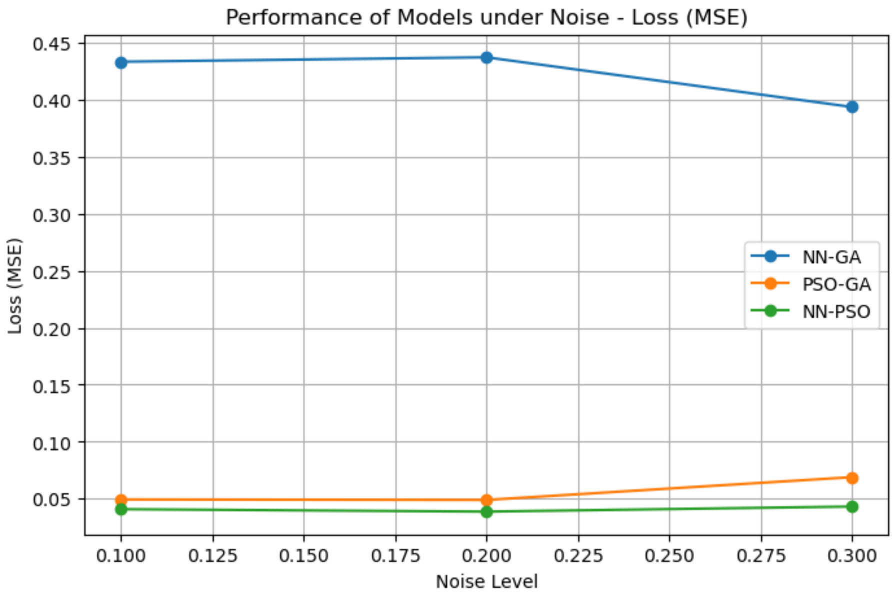

Figure 3. Performance of Models under Noise – Loss (MSE)

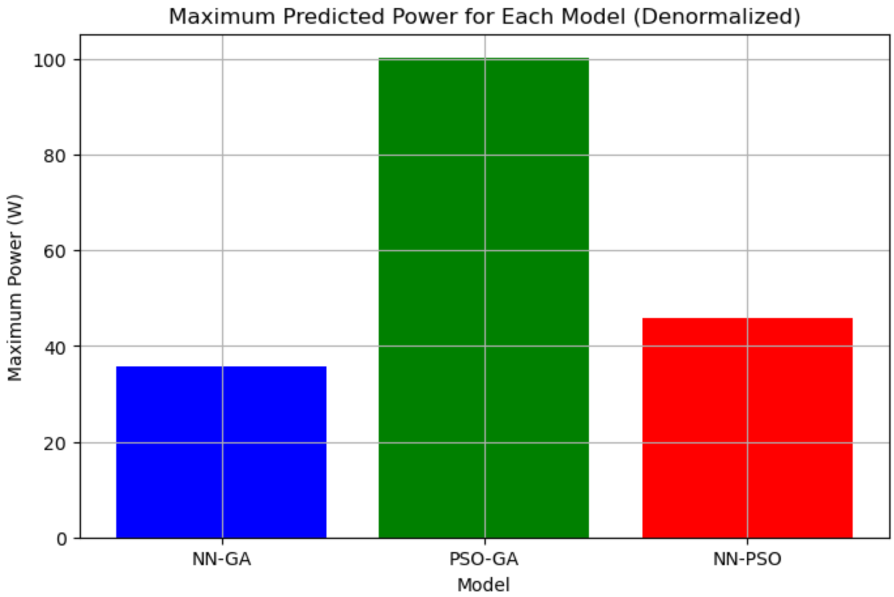

Figure 4. MPP for each model (Denormalized results)

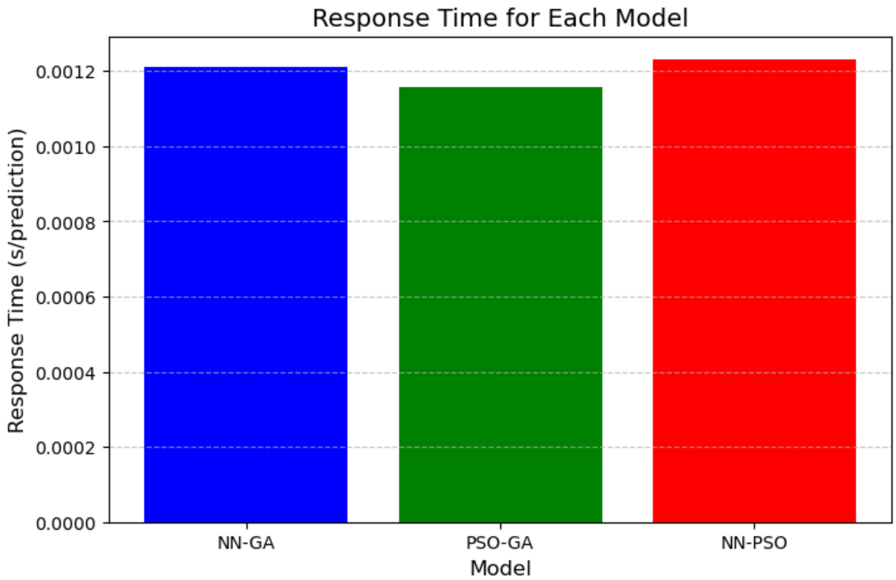

Figure 5. Response time for each model

Conclusion

The comparative evaluation of three hybrid MPPT models—NN-GA, PSO-GA, and NN-PSO—demonstrated the unique advantages and limitations of each approach. The NN-GA model excelled in minimizing response time, making it suitable for applications requiring rapid adaptation to changing conditions. However, its performance was less robust under noisy data, highlighting its limited applicability in unstable environments. The PSO-GA model achieved the highest maximum power output, demonstrating superior efficiency and robustness across all tested noise levels. This makes it an ideal choice for scenarios where power extraction is the primary objective. The NN-PSO model, while achieving intermediate results in response time and maximum power, exhibited the highest robustness to noise, as evidenced by its low MAE at varying noise levels. This balance suggests its suitability for environments characterized by fluctuating and noisy conditions.

These results underscore the importance of selecting an MPPT algorithm based on specific operational priorities, whether it be response time, power extraction efficiency, or noise robustness. Future research could explore advanced hybridization techniques, integrating additional machine learning algorithms or optimization methods to further enhance MPPT performance in diverse real-world conditions.

REFERENCES

1. Motahhir S., El Hammoumi A., El Ghzizal A. The most used MPPT algorithms: Review and the suitable low-cost embedded board for each algorithm. Journal of Cleaner Production, 2020, №246, 118983.
2. Eltawil M.A., Zhao Z. MPPT techniques for photovoltaic applications. Renewable and Sustainable Energy Reviews, 2013, №25, pp. 793-813.
3. Sarvi M., Azadian A. A comprehensive review and classified comparison of MPPT algorithms in PV systems. Energy Systems, 2022, №13(2), pp. 281-320.
4. Christopher I.W., Ramesh R. Comparative study of P&O and InC MPPT algorithms. American Journal of Engineering Research (AJER), 2013, №2(12), pp. 402-408.
5. Elgendy M.A., Zahawi B., Atkinson D.J. Evaluation of perturb and observe MPPT algorithm implementation techniques. 6th IET International Conference on Power Electronics, Machines and Drives (PEMD 2012), 2012, pp. 1-6.
6. Bendib B., Belmili H., Krim F. A survey of the most used MPPT methods: Conventional and advanced algorithms applied for photovoltaic systems. Renewable and Sustainable Energy Reviews, 2015, №45, pp. 637-648.
7. Elgendy M.A., Zahawi B., Atkinson D.J. Assessment of perturb and observe MPPT algorithm implementation techniques for PV pumping applications. IEEE Transactions on Sustainable Energy, 2011, №3(1), pp. 21-33.
8. Grossi E., Buscema M. Introduction to artificial neural networks. European Journal of Gastroenterology & Hepatology, 2007, №19(12), pp. 1046-1054.
9. Lamb A. A brief introduction to generative models. arXiv preprint arXiv:2103.00265, 2021.
10. Settles M. An introduction to particle swarm optimization. Department of Computer Science, University of Idaho, 2005, №2, pp. 12.

Accepted: 13.12.2024