STU
SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA

Journal of Electrical Engineering, Vol. 77, No. 5, 2026, pp. 585-598
https://reference-global.com/journal/JEE

A neural net fitting-based approach for optimal power flow

Jordan Radosavljević[1], Aleksandra Grujić[2], Miloš Milovanović[1]

Optimal power flow (OPF) is one of the most important management actions to achieve economical, safe and reliable operation of power systems. Require both high accuracy and speed of calculation so that appropriate corrective control actions can be implemented in real time. Classical methods such as interior-point method, sequential programming methods, as well as metaheuristic optimization methods used today require a lot of time to achieve satisfactory accuracy of OPF results. With the development of artificial intelligence and neural networks, it is possible to drastically reduce the OPF calculation time. This paper proposes the application of Neural Net Fitting (NNF) based approach for OPF. The basic idea of this work is to use a combination of metaheuristic optimization methods (MOA) and the NNF technique to realize a kind of hybrid method-network, which would solve the OPF problem in a very efficient and fast way, in a way suitable for real-time application. The proposed approach is tested on the standard IEEE 30-bus test system. The simulation results show that the application of the proposed NNF technique yields the same accuracy OPF results as the application of the classical metaheuristic methods PSOGSA or GA with the hybrid *fmincon* interior-point function in MATLAB®, but with significantly less computation time.

Keywords: artificial intelligence, optimal power flow, metaheuristic optimization methods, neural net fitting

1 Introduction

Optimal power flow is the most important management action that is implemented in real time in order to achieve the economic and desired mode of operation of the power system in accordance with the set goal.

The standard AC OPF implies that there is a set of the so-called control variables, such as the active powers of the generators, the voltage modules of the generators, the transmission ratios of the regulating transformers, and the reactive powers of the VAR compensators, the values of which can be independently changed within a certain range and thus directly affect the state of the system in economic, technical, and environmental aspects.

In essence, solving the OPF problem is finding such a combination of control variables that will give the optimal-desired state of the system for a given level of power consumption, while satisfying all technical constraints.

It is a very complex, nonlinear, dynamic optimization problem with a large number of constraints, for the solution of which different groups of methods and specialized software are used.

Metaheuristic optimization methods have proven to be very practical and robust for solving the OPF problem [1]. Their main advantage is that they can be easily applied to the most complex OPF problems, able to give the optimal or near-optimal solution. Unlike classical methods, they do not require any simplification and linearization of the OPF mathematical model. They can take into account various forms of constraints and objective functions, which do not have to be differentiable. However, their main drawback is the relatively long calculation time, which for real systems of large dimensions is a major limiting factor for real-time application. Accordingly, with the development of powerful computers, specialized methods and techniques were developed to speed up the OPF calculation process as much as possible, while maintaining the necessary calculation accuracy.

A bibliometric analysis [2] shows that traditional deep learning techniques like long short-term memory (LSTMs) and convolutional neural networks (CNNs) are widely used in power systems. In recent research methods like advanced reinforcement learning (RL), graph neural networks (GNNs), and physics-informed neural networks (PINNs) are emerging, which have proven to be very suitable for effectively solving the OPF problem.

Nelikat and Chachivasilejadis [3] were the first to propose PINNs for accurate estimation of AC OPF results. They presented a framework for incorporating AC OPF equations into neural network training. They

[1]University of Priština in Kosovska Mitrovica, Faculty of Technical Sciences, Kosovska Mitrovica, Serbia
[2]Academy of Technical and Art Applied Studies, School of Electrical and Computer Engineering, Belgrade, Serbia
jordan.radosavljevic@pr.ac.rs

https://doi.org/10.2478/jee-2026-0055, Print (till 2015) ISSN 1335-3632, On-line ISSN 1339-309X
© This is an open access article licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives License (http://creativecommons.org/licenses/by-nc-nd/4.0/).

also proposed an approach based on worst-case guarantees for determining and reducing constraint violations in the OPF problem.

The authors in [4] have developed a deep neural network (DNN)-based method to accelerate the achievement global optimal OPF solution. The DNN model learn the mapping between the power loads and the buses and branches. Then, the optimization-based bound tightening (OBBT) is used to solve the AC OPF problem ensuring satisfying all technical constraints of the OPF problem.

Lotfi and Pirnia [5] proposed a DNN algorithm to classify loads, and then to predict optimal results of the OPF problem for each feasible pair of active and reactive power of loads. The proposed method enables Estimations of the OPF results with high accuracy and fast run time.

The DeepOPF approach [6-8] demonstrated that large speedups are possible on DC-OPF and AC-OPF variants, with subsequent papers improving feasibility, handling security constraints, and enabling flexible topology. Further application of DeepOPF is proposed in [9]. First, the DNN is trained to predict a set of control variables, and then the remaining unknown variables are directly calculated by solving the power flow equations. Deep RL (DRL) algorithm have been proposed in [10, 11] for achieving fast AC OPF solutions. The proposed method in [10] adopts imitation learning (IL) and in [11] a supervised-learning method to generate good initial weight for the neural networks, whereas both works use a proximal policy optimization (PPO) algorithm to train and test agents. Both approaches were tested on the Illinois 200-bus system. Near-optimal OPF solutions are achieved at least 7 times speedup compared with the interior-point method in [10], whereas for the proposed methodology in [8] this speedup is 14 times.

Tran et al [12] proposed a combination of CNNs and a self-attention mechanism to rearranging the power network parameters in an image-like shape of a multi-channel image where each channel is a two-dimensional matrix. Densely connected network – DenseNet integrated with a self-attention mechanism – abbreviated as DeAtNet is used to map a rule between the information of the power system and optimal solution of the OPF problem.

An improved proximal policy optimization (PPO) algo-rithm for sequential security-constrained OPF (SCOPF) based on expert knowledge and safety layer is proposed in [13]. The proposed method allows safe, stable, and economic operation of the power system. A study presented in [14] combines particle swarm optimization (PSO) algorithm with artificial neural networks (ANN) to estimate optimal solution of the OPF problem. The ANN is used here to determining the optimal parameters of PSO algorithm for the optimization problem.

The GNN treats the power system as a graph to learn node-level decision while respecting the network structure. One of the first applications of GNN for OPF is proposed by authors in [15]. The authors used a parametrized GNN model, which is local and scalable by design, and trained to imitate the optimal solution obtained using an interior point solver. Once trained, the GNN offers an efficient estimation of AC OPF solutions.

A new solution methodology for the OPF problem by applying typed GNNs is presented in [16]. This approach allows the representation of different elements of power system with different types of nodes, thus adding accuracy and interpretability to the solutions obtained in comparison to results of conventional feed-forward neural networks models.

Sun et al [17] proposed a look-ahead dispatch of power systems based on linear AC OPF framework with nonlinear frequency constraints using a PINN. The PINN approach provides prediction of key frequency-control parameter values, i.e., virtual inertia and droop coefficients, accurately.

A novel physics-guided safe policy learning method is proposed in [18], featuring an analytical evaluation model to provide both accurate safety and cost-efficiency evaluations in real-time implementation, for solving dynamic security-constrained OPF (DSCOPF).

The authors in [19] also propose a scalable PINN model that leverages the inclusion of physical constraints into the loss function using a penalty factor and the utilization of bounds of optimization variables in the activation functions to enhance the generalization performance of tuned neural networks. The simulation results show that the proposed approach significantly improves the success rate and computational speed in solving the OPF problem, especially when predictions are used as warm start points.

Another PINN-based model for OPF in power systems with high penetration of renewable energy sources (RES) is proposed in [20]. In this work, the network′s physics is represented by AC OPF constraints, using the Karush-Kuhn-Tucker (KKT) conditions.

Wu et al [21] have introduced a primal-dual approach to learn optimal constrained deep reinforce-ment learning (DRL) policies specifically for predictive control in real-time stochastic dynamic OPF. The control variables in the dynamic OPF are power generations and battery outputs while satisfying critical constraints.

The variability nature of RES generation and the different operating conditions of power system necessitate frequent neural network updates with new data

instances. The authors in [22] proposed a semi-supervised learning framework aided by data augmentation to address this need and reduce the time required for data preparation time. They used ridge regression instead the traditional solver, facilitating swift prediction of optimal solutions for the given input load demands.

To address constraint violations and guarantee feasibility, many papers combine a learned NN predictor with a post-processing optimization or projection step [23, 24]. This hybrid strategy trades some speed for more reliable feasibility and smaller optimality gaps, and is practical for operator deployment.

This paper introduces an NNF-based framework for solving the OPF problem. The core concept is to integrate MOAs with the NNF methodology to form a hybrid optimization-learning model capable of delivering OPF solutions with high efficiency and significantly reduced computation time. The proposed approach is designed to support real-time or near-real-time operational requirements. Due to the robustness of MOA for solving OPF problems, it is possible to train NNF to the most complex OPF problems with different objective functions and constraints.

The main contributions of this can be summarized as follows:

- Specific comparison of MOA to get the best metaheuristic method for NNF training.
- Using MOA to train NNF for OPF evaluation.
- Definition of a fast and efficient method based on MOA-NNF for real-time OPF computation.
- The demonstration that the proposed MOA-NNF framework can accommodate different decision variables, objective functions, and operational constraints without requiring simplification or linearization of the original OPF mathematical formulation.

2 OPF problem formulation

The OPF problem can be expressed mathematically as follows [1]:

$$\text{optimize } F(\boldsymbol{x}, \boldsymbol{y}) \tag{1}$$

Optimization, i.e., minimization or maximization, of the objective function F is carried out while satisfying certain constraints:

$$g(\boldsymbol{x}, \boldsymbol{y}) = 0 \tag{2}$$

$$h(\boldsymbol{x}, \boldsymbol{y}) \leq 0 \tag{3}$$

$$\boldsymbol{x} \in \boldsymbol{X} \tag{4}$$

In the previous equations, $\boldsymbol{x}$ is a vector of control variables, whose elements are most often generator active power outputs P_G (except at the slack bus, supposed P_{G1}), generator voltages V_G, transformer tap settings T, and shunt Var compensations Q_C. Therefore, the vector $\boldsymbol{x}$ can be expressed as follows:

$$\boldsymbol{x} = [P_{G2} \dots P_{GNG}, V_{G1} \dots V_{GNG}, T_1 \dots T_{NT}, Q_{C1} \dots Q_{CNC}] \tag{5}$$

where NG, NT and NC are the number of generators, regulating transformers and VAR compensators, respectively.

$\boldsymbol{y}$ denotes the vector of dependent variables in the OPF problem. The elements of this vector are the slack bus power P_{Gsl}, load bus voltages V_L, generator reactive power outputs Q_G, and transmission line loadings S_l:

$$\boldsymbol{y} = [P_{Gsl},\ V_{L1} \dots V_{LNL}, Q_{G1} \dots Q_{GNG}, S_{l1} \dots S_{lNTL}] \tag{6}$$

where NL and NTL are number of load buses and transmission lines, respectively.

2.1 Equality constraints

The equality constraints (2) are the equations of AC power flow, which ensure the achievement of the balance of active and reactive powers in the system.

$$P_{Gi} - P_{Di} - V_i \sum_{j=1}^{NB} V_j \left(G_{ij} \cos\theta_{ij} + B_{ij} \sin\theta_{ij} \right) = 0 \tag{7}$$

$$Q_{Gi} - Q_{Di} - V_i \sum_{j=1}^{NB} V_j \left(G_{ij} \sin\theta_{ij} - B_{ij} \cos\theta_{ij} \right) = 0 \tag{8}$$

where i=1, …, NB; NB is the number of busses, P_{Gi} is the active power generation, Q_{Gi} is the reactive power generation, P_{Di} is the active load demand, Q_{Di} is the reactive load demand, θ_{ij} is voltage angle between busses i and j, G_{ij} and B_{ij} are the real and imaginary terms of bus admittance matrix corresponding to ith row and jth column, respectively.

2.2 Inequality constraints

Each OPF solution must be such that the voltage magnitude at the load buses and the reactive powers of generators are within their predefined limits (min-max). Also, the power flows, or currents, on all branches of the network must be such that the thermal limitation of the system elements is not violated. The mentioned constraints are the so-called functional operating constraints of the OPF problem (3), and can be expressed as follows:

$$V_{Li}^{min} \leq V_{Li} \leq V_{Li}^{max}, \quad i=1, \dots, NL \tag{9}$$

$$Q_{Gi}^{min} \leq Q_{Gi} \leq Q_{Gi}^{max}, \quad i=1, \dots, NG \tag{10}$$

$$S_{li} \leq S_{li}^{max}, \quad i=1, \dots, NTL \tag{11}$$

Constraints (9) and (11) are also known as the security constraints.

The solution of the OPF problem is located in a certain space of possible solutions. This space of possible solutions of the OPF problem is defined by the possible values of the control variables, which are located inside the predefined limits of these variables. Constraints (4), therefore, defines the space of possible solutions of the OPF problem, and can be expressed as follows.

$$P_{Gi}^{min} \le P_{Gi} \le P_{Gi}^{max}, \quad i=1, \ldots, NG \tag{12}$$

$$V_{Gi}^{min} \le V_{Gi} \le V_{Gi}^{max}, \quad i=1, \ldots, NG \tag{13}$$

$$T_{i}^{min} \le T_{i} \le T_{i}^{max}, \quad i=1, \ldots, NT \tag{14}$$

$$Q_{Ci}^{min} \le Q_{Ci} \le Q_{Ci}^{max}, \quad i=1, \ldots, NC \tag{15}$$

.

2.3 Objective function

The OPF problem can be considered for different objective functions. The basic objective function for the OPF problem (1) is to minimize the total operating cost of generators in the system, *Fcost,* where the generator cost characteristic *f* is defined as a quadratic function of the generator output power P_G:

$$\min_{\boldsymbol{x}} F(\boldsymbol{x},\boldsymbol{y}) = \min_{\boldsymbol{x}} Fcost(\boldsymbol{x},\boldsymbol{y}) = \min_{\boldsymbol{x}} \sum_{i=1}^{NG} f_i(P_{Gi}) = \min_{\boldsymbol{x}} \sum_{i=1}^{NG} \left(a_i + b_i P_{Gi} + c_i P_{Gi}^2\right) \tag{16}$$

where a_i, b_i and c_i are the cost coefficients of the *i*-th generator, and P_{Gi} is the corresponding active power output.

In practice, for large generators in thermal power plants, it is necessary to take into account the effect of sequential opening of steam turbine valves to regulate the active power output. This effect, known as the valve-point effect, introduces an additional nonlinearity into the generator cost function, in the form of a repetitively rectified sine function superimposed on the basic quadratic characteristic of the generator, which can be expressed as follows:

$$\min_{\boldsymbol{x}} F(\boldsymbol{x},\boldsymbol{y}) = \min_{\boldsymbol{x}} Fcostvpe(\boldsymbol{x},\boldsymbol{y}) = \min_{\boldsymbol{x}} \sum_{i=1}^{NG} a_i + b_i P_{Gi} + c_i P_{Gi}^2 + \left| d_i \sin\left(e_i\left(P_{Gi}^{min} - P_{Gi}\right)\right)\right| \tag{17}$$

where d_i and e_i are fuel cost coefficients for valve-point effects.

It should be emphasized that the control variables defined by the vector $\boldsymbol{x}$ are self-constrained, because they determine the space of possible solutions to the OPF problem. However, for dependent variables represented by the vector $\boldsymbol{y}$, i.e., the active power generation at the slack bus P_{Gsl}, load bus voltages V_L, reactive power generations Q_G, and branch apparent power flows S_l, an additional mechanism needs to ensure that the values of these variables be within the given constraints. Only OPF solutions that satisfy all the defined constraints can be considered valid. One of the common approaches for that, which was also used here, is the expansion of the objective function by adding the appropriate quadratic penalty terms for each dependent variable in vector $\boldsymbol{y}$ [25, 26]. The new expanded objective function to be optimized becomes:

$$F_p(\boldsymbol{x},\boldsymbol{y}) = F(\boldsymbol{x},\boldsymbol{y}) + \lambda_p\left(P_{Gsl} - P_{Gsl}^{lim}\right)^2 + \lambda_V \sum_{i=1}^{NL}\left(V_{Li} - V_{Li}^{lim}\right)^2 + \lambda_Q \sum_{i=1}^{NG}\left(Q_{Gi} - Q_{Gi}^{lim}\right)^2 + \lambda_S \sum_{i=1}^{NTL}\left(S_{li} - S_{li}^{lim}\right)^2 \tag{18}$$

where λ_P, λ_V, λ_Q and λ_S are the penalty factors, y^{lim} is the limit value of the dependent variable *y* and is given as:

$$y^{lim} = y^{max} \text{ if } y > y^{max}$$
$$y^{lim} = y^{min} \text{ if } y < y^{min} \tag{19}$$

3 Methodology

In practice, the OPF problem is solved several times a day in periods ranging from a few minutes to tens of minutes, with only the level of load usually changing, while the values of other parameters such as transmission line parameters, transformers, thermal constraints, generator constraints, and transmission network topology, remain unchanged [4]. This implies the possibility of applying artificial intelligence, namely NNF-based approach, to estimate the OPF solution.

The basic idea of this work is to use a combination of metaheuristic optimization methods (MOA) and the neural net fitting (NNF) technique to realize a kind of hybrid method-network, which would solve the OPF problem in a very efficient and fast way, in a way suitable for real-time application.

It is based on the fact that from an arbitrary number of proven MOAs, the "best" method for NNF training is selected based on a statistical evaluation of the obtained OPF solutions in the test system of interest. Then, by applying these methods, the OPF is calculated for a certain relevant number, say 2000 randomly generated power consumptions in a certain range (min-max), according to the minimum and maximum power from the load diagram in a given system. It is assumed that the load diagrams and power factor are the same in all consumer nodes of the system.

The obtained MOA OPF results – the optimal values of the control variables and the corresponding load data – are now used for training the NNF network for OPF. This work uses the NNF application in MATLAB®.

The NNF application solve fitting problem using two-layer feed-forward networks, as shown in Fig. 1. This application provides built-in training algorithms that can be used to train a neural network.

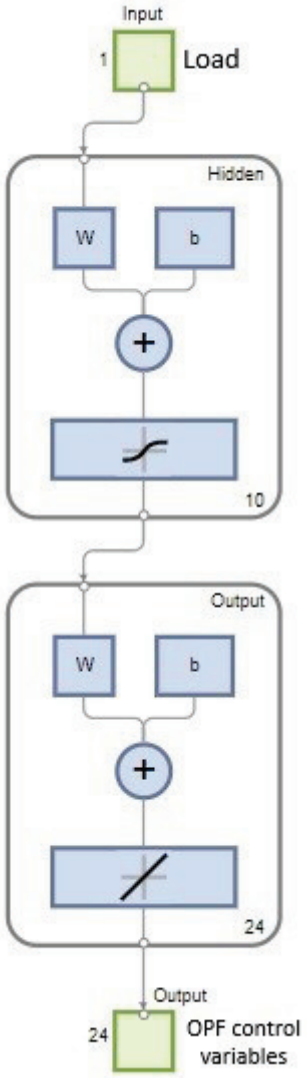


Fig. 1. Training of NNF for OPF

First, data is imported from the MATLAB® workspace. After that, this data is divided into training, validation, and testing sets. Then, the network parameters are defined, i.e., the percentage of data for training, validation, and testing, the choice of the algorithm training method, and then the neural network training process is started. After the network training process is completed, the NNF application allows for the evaluation of the network performance using mean square error analysis and regression analysis. The results can also be analyzed using visualization graphs, such as regression analysis or error histogram. A MATLAB® script file is also generated to the trained NNF very fast evaluate OPF results for any input load data.

Levenberg-Marquardt training algorithm was used in this work. This algorithm is often the fastest training algorithm, although it does require more memory than other techniques. For its implementation, the NNF application in MATLAB® uses the *trainlm* function. The training of the network is done off-line, and the obtained MOA-NNF network is used for online OPF calculation.

The proposed methodology is briefly described below, step by step:

1. Defining complete data on the power system (test system) for the calculation of OPF. This implies all parameters and variables necessary for the application of the mathematical model OPF according to Eqns. (1)-(19).
2. OPF calculation using different metaheuristic methods and statistical evaluation of the obtained results. Choosing the best method, as illustrated in Fig. 2.

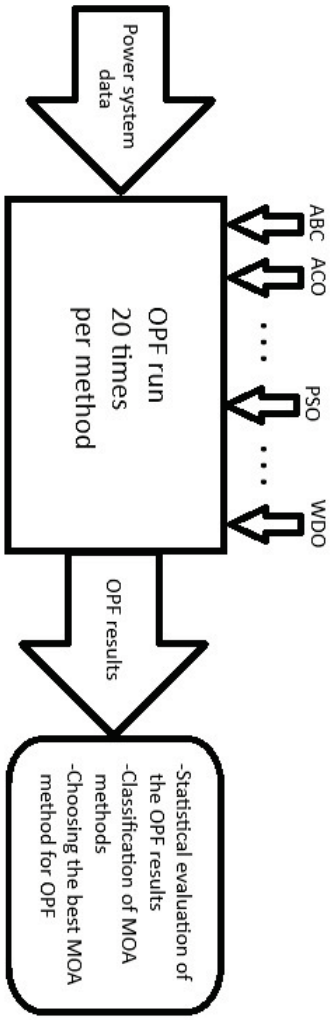


Fig. 2. Choosing the best MOA for OPF

3. Generating a large number (e.g., 2000) of random load power levels in a given range (min-max).
4. Generating data for training the neural network, i.e., calculating the OPF for all 2000 load power levels. The obtained OPF results are the optimal values of the control variables for all 2000 load levels, as in Fig. 3.

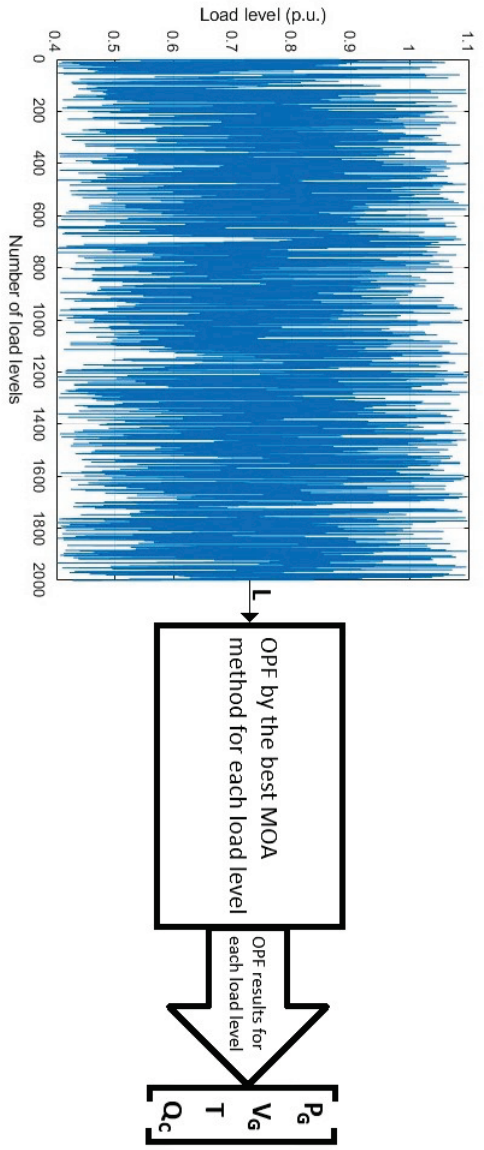


Fig. 3. Generating data for training the NNF

5. Training the NNF with the obtained OPF data is performed using the MATLAB© application Neural net fitting as follows:
   - Splitting the obtained OPF data into training, validation, and test sets.
   - Defining and training the NNF.
   - Evaluation of the network performance using mean squared error and regression analysis.
   - Analyzing of the results using visualization plots, such as regression fit or histogram of errors.
   - Generate MATLAB scripts to reproduce OPF results and customize the training process.
6. Application of the obtained NNF script to solve OPF. Testing for arbitrarily chosen load levels and comparison of results.

   The described steps are illustrated in Fig. 4.

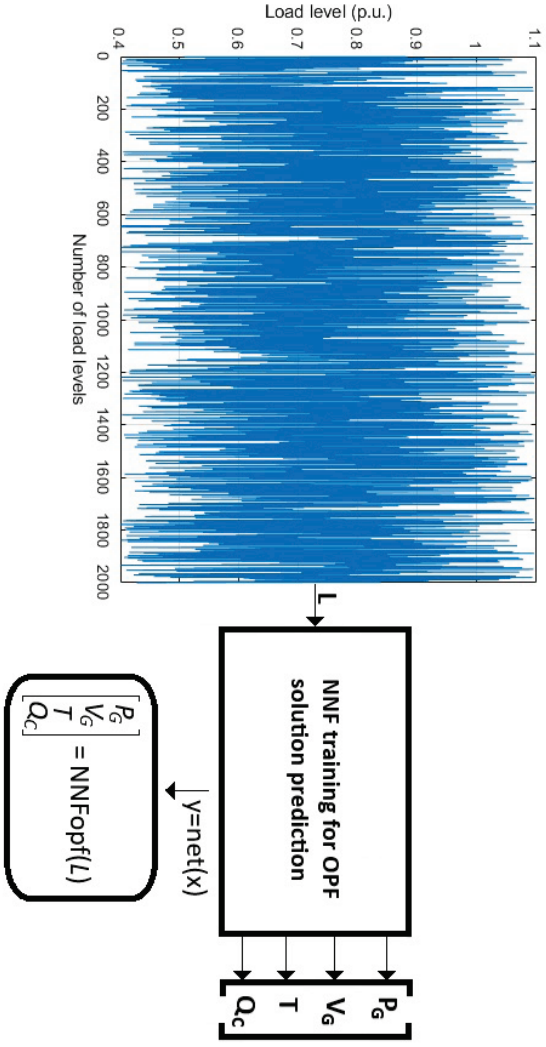


Fig. 4. Training the NNF for OPF

4 Simulation results

Testing of the proposed methodology was performed on the standard IEEE 30-bus test system according to the procedure described in the previous Section 3.

In the literature, this is the standard test system for OPF results evaluation. It is very useful for a fair comparison of different OPF methods. The single-line diagram of the IEEE 30-bus test system is shown in Fig. 5. There are 24 OPF control variables with noted lower and upperlimits. The complete system data can be found in [1]. The voltages of all load bus have been constrained within limits of 0.95 and 1.05 p.u.

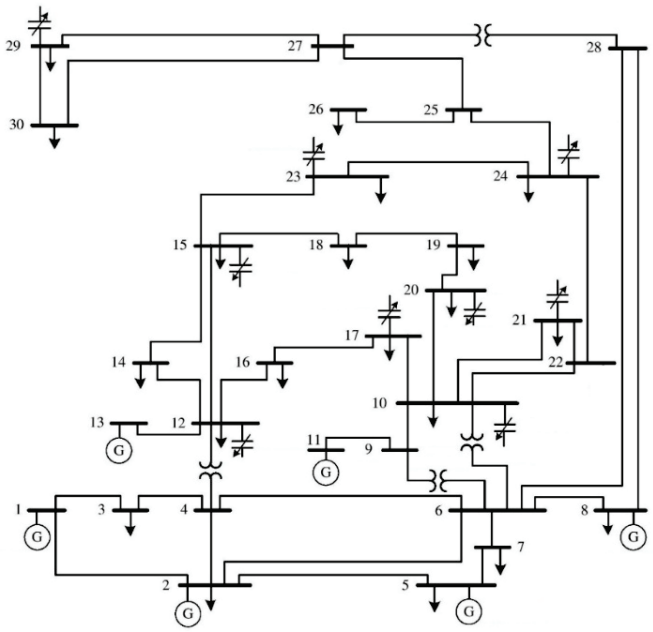


Fig. 5. IEEE 30-bus test system [1]

4.1 Choosing the best MOA for OPF

In this paper, 24 different MOAs were considered, as proposed in [1], namely artificial bee colony (ABC), ant colony optimization (ACO), ant lion optimizer (ALO), biogeography-based optimization (BBO), backtracking search optimization algorithm (BSA), cuckoo search (CS), differential evolution (DE), equilibrium optimizer (EO), firefly algorithm (FFA), gradient-based optimizer (GBO), grey wolf optimizer (GWO), harmony search (HS), krill herd algorithm (KH), moth-flame optimization algorithm (MFO), moth swarm algorithm (MSA), political optimizer (PO), PSO, gravitational search algorithm (GSA), hybrid of PSO and GSA (PSOGSA), Runge Kutta optimizer (RUN), swarm robotics search & rescue (SRSR), teaching-learning-based optimization (TLBO), tunicate swarm algorithm (TSA), and wind-driven optimization (WDO).

A detailed description of these methods can be found in [1]. The algorithmic parameters of these methods, when applied to the calculation of OPF, were taken as proposed by their authors.

The comparison of these methods in solving the OPF problem is performed on the IEEE 30-bus test system, with the same objective function, control variables, and constraints. A load level of 0.7585 p.u. was adopted randomly within the range of 0.4 to 1.1 p.u.

The statistical indicators obtained over 20 runs for each of the optimization methods are shown in Tab. 1. The OPF calculation was performed for the objective function of minimizing the generator costs. The population size of 50 and max iteration number of 200 are the same for each algorithm.

Table 1. OPF results for the load level L=0.7585 (p.u.)

	Min	Max	Mean	Std.	TOC mean (s)
ABC	565.20	565.91	565.42	0.17	32.01
ACO	567.59	587.72	574.71	5.08	15.97
ALO	565.38	567.31	565.94	0.61	19.27
BBO	568.20	581.76	572.99	3.30	15.65
BSA	565.76	567.36	566.36	0.43	15.81
CS	565.43	566.28	565.76	0.23	16.37
DE	565.30	565.64	565.44	0.07	22.55
EO	565.17	565.69	565.35	0.17	18.29
FFA	566.97	583.16	571.71	4.69	15.71
GBO	565.15	565.40	565.22	0.06	17.37
GWO	568.66	597.96	580.76	9.21	16.89
HS	565.39	566.67	566.00	0.35	6.55
KH	571.86	589.14	581.30	5.48	16.12
MFO	565.19	573.21	566.04	1.76	37.69
MSA	565.20	565.97	565.46	0.20	16.63
PO	565.19	578.61	567.36	3.06	17.25
PSO	565.37	639.53	572.62	17.94	16.23
GSA	565.58	568.10	566.37	0.63	16.21
PSOGSA*	565.14	565.54	565.21	0.08	17.00
RUN	565.24	565.48	565.39	0.05	29.97
SRSR	566.75	567.87	567.40	0.30	32.39
TLBO	565.16	565.40	565.22	0.06	36.19
TSA	567.08	603.59	582.78	10.50	15.88
WDO	565.46	566.76	565.72	0.34	15.57

By analysing the results in Tab. 1, it can be seen that several algorithms, such as DE, GBO, PSOGSA, RUN, and TLBO, provide high-quality and stable solutions to the OPF problem. The results obtained are consistent with the considered objective function and all the stated constraints are met. Taking into account all statistical indicators, it can be said that the winner in this small competition in solving the OPF problem is the PSOGSA method, and as such it will be further used for training the NNF network for OPF.

4.2 Generating data for training the NNF

First, a large number (in this case 2000) of random values of the load power level in the range from 0.4 p.u. to 1.1 p.u. is generated, relative to the nominal load power in the system nodes. The results of generating these powers are given in Fig. 6. As mentioned earlier, it

is assumed that the relative daily power diagrams and power factors are the same in all nodes, so that in each node the power is obtained by multiplying the nominal value of active and reactive power with a randomly generated relative value in the given range (0.4 p.u.-1.1 p.u.).

To provide data for training the NNT network, for each of the 2000 generated power of load levels, the OPF is calculated using the selected PSOGSA method, as illustrated in Fig. 3.

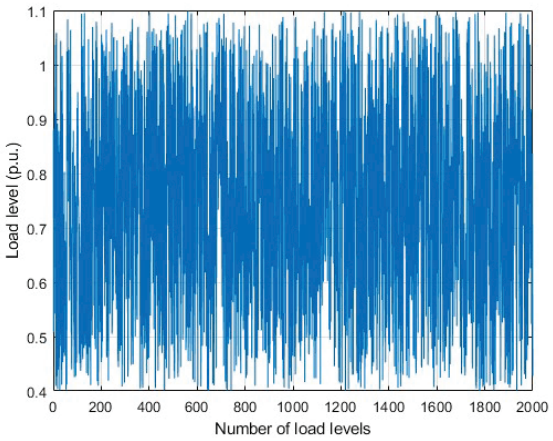


Fig. 6. Randomly generated load levels in a defined range.

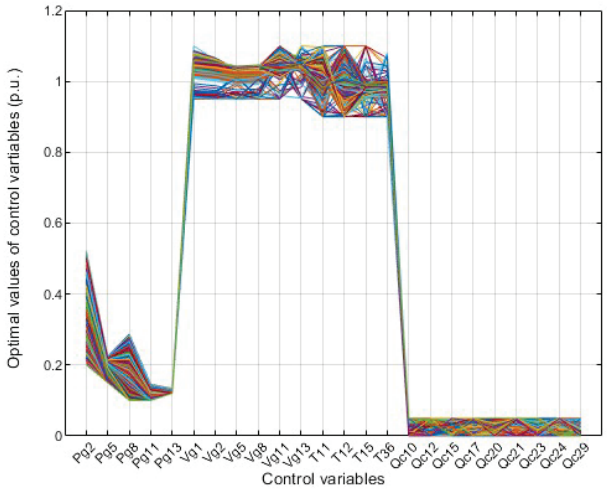


Fig. 7. The OPF results for randomly generated load levels

The OPF results - optimal values of control variables for 2000 different load levels from Fig. 6 are shown in Fig. 7. It should be noted that the optimal generator and VAR compensator powers in Fig. 7 are given in (p.u.) for a base power of 100 MVA.

4.3 Training the NNF for OPF

Using the obtained data in Figs. 6 and 7, the NNF network is trained in the MATLAB Neural Net Fitting application, as shown in Figure 4, using two-layer feed-forward networks. The training data, in Figs. 6 and 7, are divided so that 70% is used for training, and 15% each for validation and testing. The layer size is set to 10.

The Levenberg-Marquardt training algorithm was used, which updates weight and bias values according to Levenberg-Marquardt optimization. Levenberg-Marquardt training is often the fastest training algorithm, although it does require more memory than other techniques. To implement this algorithm, the NNF application in MATLAB uses the *trainlm* function.

Performances of training NNF for OPF using the data given in Figs. 6 and 7, are shown in Figs. 8-12. Figure 8 shows the function fit for output element with marked training targets, training outputs, validation targets, validation outputs, test targets, errors and fit. As can be seen, the target and outputs agree very well, which implies a high evaluation accuracy of the OPF results using the trained NNF network. Table 2 shows the amount of data for training, validation and testing, as well as the value of the mean squared deviation i.e., the mean squared difference between outputs and responses, and R-value which means the correlation between outputs and responses.

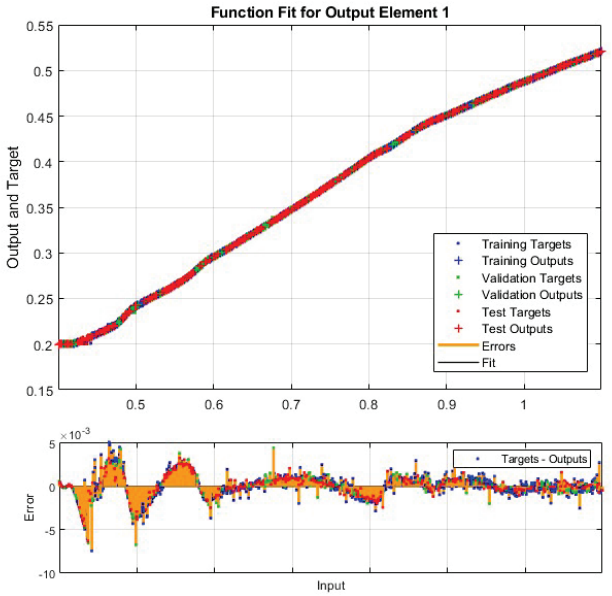


Fig. 8. Outputs and targets of NNF for OPF

Table 2. Indicators of NNF training for OPF

	Observations	MSE	*R*
Training	1400	3.77e-04	0.999
Validation	300	3.90e-04	0.999
Test	300	3.82e-04	0.999

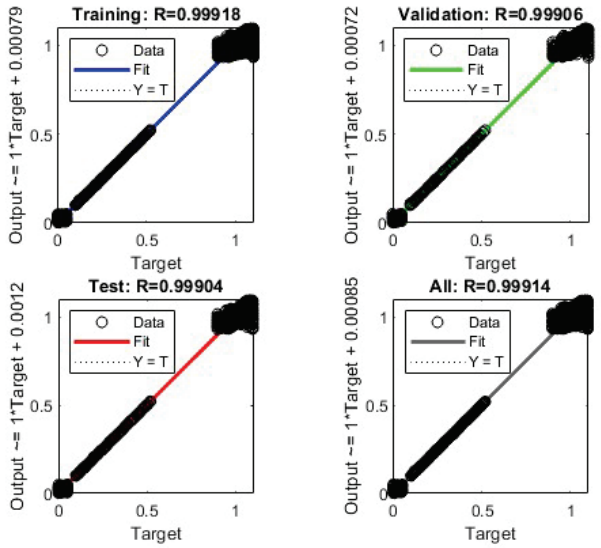


Fig. 9. Linear regression

Figure 9 shows the regression analysis of a neural network for training, validation, test, and all data. In each plot, predicted output values closely follow the ideal Y=T reference line, with correlation coefficients of R=0.999 for all data indicating excellent agreement between predicted and target values.

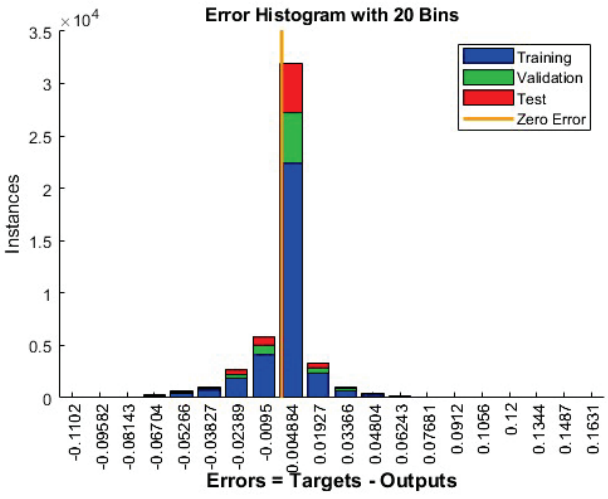


Fig. 10. Error histogram

Error histogram with 20 bins in Fig. 10 shows the distribution of prediction errors, defined as targets minus outputs, for the training, validation, and test datasets. The histogram is strongly concentrated near zero error, with the highest frequency occurring around −0.005 to 0.005. Training instances are shown in blue, validation instances in green, and test instances in red. An orange vertical line marks zero error.

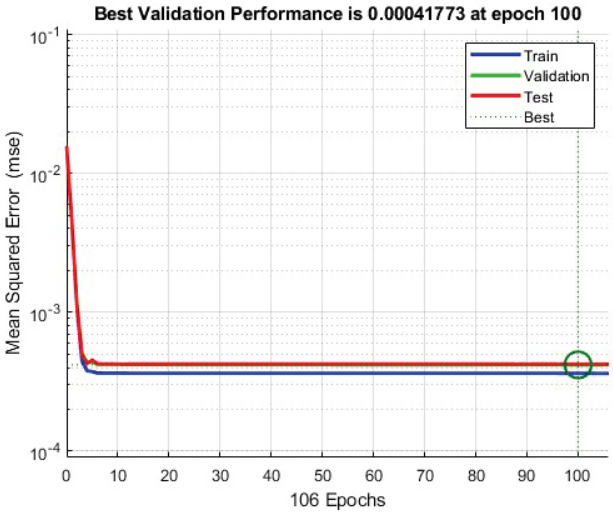


Fig. 11. Mean square error diagram.

Figure 11 shows mean squared error versus training epochs on a logarithmic y-axis. Training, validation, and test errors drop sharply during the first few epochs and then level off near 4×10^{-4}. The best validation performance is 0.00041773 at epoch 100, marked by a green dotted vertical line and a circled point.

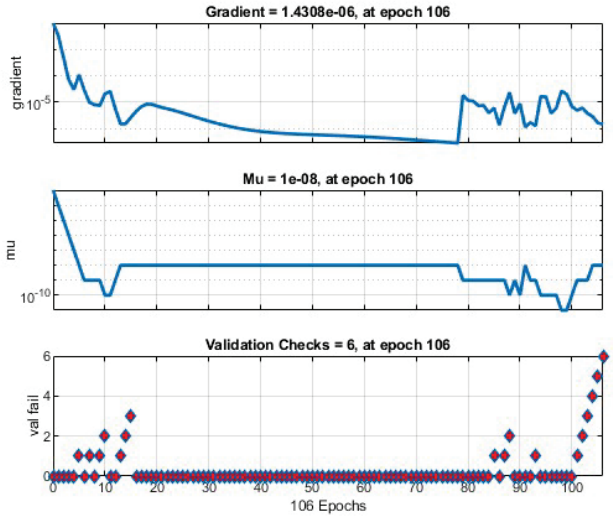


Fig. 12. Training state

Figure 12 shows training diagnostics over 106 epochs. The top plot shows the gradient decreasing from about 10^{-4} to below 10^{-5}, followed by fluctuations near 10^{-5} after epoch 80. The middle plot shows μ decreasing to about 10^{-10}, remaining nearly constant, then fluctuating and briefly dropping below 10^{-10} near epoch 98. The bottom plot shows validation-check failures, mostly at zero, with occasional failures early in training and increasing failures after epoch 80, reaching 6 at epoch 106.

4.4 Verification NNF for OPF

The trained NNF network was evaluated on two operating conditions: the nominal load L=1 p.u. and a reduced load of L=0.5 p.u. The corresponding OPF results are summarized in Tabs. 3 and 4, respectively. The objective was to minimize total generation fuel costs, modeled as quadratic functions of generator output. For verification, OPF was also solved using the PSOGSA method and, additionally, with a hybrid GA-*fmincon* approach employing the interior-point algorithm, which served as a reference method to validate the accuracy of the obtained solutions.

As can be seen, all three methods give practically the same OPF results for both load levels, but the computation time for NNF is drastically shorter than for PSOGSA and GA-*fmincon* methods. This difference shows that with the NNF approach, OPF results for a given load level are obtained practically instantaneously, which makes it very suitable for real-time use.

Compared to the initial (base) case without OPF, with using OPF algorithms for a load level of 1 p.u. significant fuel cost savings are achieved with approximately the same value of 11.25% for the PSOGSA algorithm, the proposed NNF method and the GA-*fmincon* method. Table 3 also gives the corresponding total active power losses in the application of individual optimization methods for minimizing fuel costs *Fcost*. As can be seen, these values are very close for all three methods, which indicates in favor of the quality of the NNF method.

Applying OPF for a load level of 0.5 also leads to a significant reduction in fuel costs of 39.59%, which is approximately the same for all three methods.

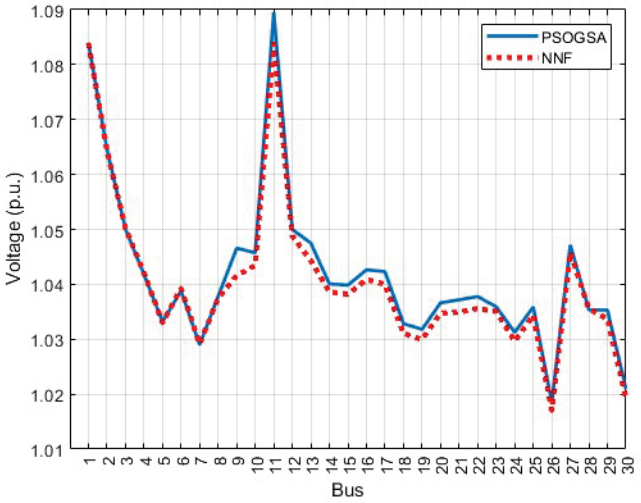


Fig. 13. Voltage profile in IEEE30-bus test system for load level of 1 (p.u.)

Figure 13 shows the comparative voltage profiles for the optimal results obtained by applying the PSOGSA and NNF methods. As can be seen, all voltages are within the defined limits. The voltages at all load nodes are within the permissible limits of (0.95, 1.05) p.u. for both methods. This indicates that the NNF-based solution produces a feasible and acceptable voltage profile in the IEEE 30-bus test system, with no violations of the allowed voltage range. The fact that the NNF prediction consistently respects these limits further confirms its suitability for real-time OPF applications and demonstrates that the NNF has effectively learned the IEEE 30-bus test system's operational constraints.

The OPF results for the objective function considering valve-point effect is presented in Tabs. 5 and 6. This calculation was carried out in order to demonstrate the effectiveness of the proposed methodology and for complex objective functions, such as the fuel cost function with the valve-point effect. In this case the OPF was performed for the nominal load L=1 p.u., and for 75% of the nominal load, L=0.75 p.u.

Similar to the previous case for the quadratic fuel cost function in Tabs. 3 and 4, it can be seen that all three methods give very similar OPF results for both load levels, but the computation time for NNF is drastically shorter than for PSOGSA and GA-*fmincon* methods.

Considering the valve-point effect, using OPF algorithms for a load level of 1 p.u., significant fuel cost savings are achieved compared to the initial (base) case, namely 14.17% using the PSOGSA algorithm, the proposed NNF method and using the GA-*fmincon* method. For a load level of 0.75, the PSOGSA method achieves fuel cost savings of 26.75%, NNF approach achives 26.63% savings, and the GA-*fmincon* 26.75%. As can be seen, the results are very close here, which also in this case confirms the quality of the proposed NNF approach for OPF.

Tables 5 and 6 also give the corresponding total active power losses for different optimization methods. As can be seen, these values are very close for all three methods, which indicates in favor of the quality of the NNF method.

Table 3. OPF results for *Fcost* minimization in the IEEE 30-bus test system for load level of 1 (p.u.)

	Initial	PSOGSA	NNF	GA-*fmincon*
P_{G1}(MW)	99.22	177.16	176.98	177.12
P_{G2}(MW)	80	48.67	48.78	48.72
P_{G5}(MW)	50	21.39	21.40	21.38
P_{G8}(MW)	20	21.27	21.36	21.26
P_{G11} (MW)	20	11.94	11.98	11.94
P_{G13}(MW)	20	12.00	11.93	12.00
V_{G1}(p.u.)	1.05	1.08	1.08	1.08
V_{G2}(p.u.)	1.04	1.06	1.06	1.06
V_{G5}(p.u.)	1.01	1.03	1.03	1.03
V_{G8}(p.u.)	1.01	1.04	1.04	1.04
V_{G11}(p.u.)	1.05	1.09	1.08	1.09
V_{G13}(p.u.)	1.05	1.05	1.04	1.04
T_{11}(p.u.)	1.08	1.03	1.04	1.04
T_{12}(p.u.)	1.07	0.94	0.95	0.94
T_{15}(p.u.)	1.03	0.97	0.97	0.96
T_{36}(p.u.)	1.07	0.97	0.97	0.97
Q_{c10} (MVAr)	0	0.02	3.36	0.12
Q_{c12}(MVAr)	0	0.04	2.38	2.97
Q_{c15}(MVAr)	0	4.94	4.08	4.46
Q_{c17}(MVAr)	0	5.00	4.70	4.99
Q_{c20}(MVAr)	0	4.24	4.42	4.24
Q_{c21}(MVAr)	0	4.99	4.91	4.99
Q_{c23}(MVAr)	0	2.98	3.72	3.24
Q_{c24}(MVAr)	0	5.00	4.96	4.99
Q_{c29}(MVAr)	0	2.59	2.50	2.53
Fcost($/h)	901.95	800.52	800.52	800.52
Ploss (MW)	5.82	9.03	9.02	9.03
TOC (s)	-	19.74	0.03	34.59

Table 4. OPF results for *Fcost* minimization in the IEEE 30-bus test system for load level of 0.5 (p.u.)

	Initial	PSOGSA	NNF	GA-*fmincon*
P_{G1}(MW)	–46.61	72.39	72.79	72.56
P_{G2}(MW)	80	24.12	23.64	23.85
P_{G5}(MW)	50	15.00	14.99	15.00
P_{G8}(MW)	20	10.00	9.99	10.00
P_{G11} (MW)	20	10.00	9.99	10.00
P_{G13}(MW)	20	12.00	11.99	12.00
V_{G1}(p.u.)	1.05	1.04	1.06	1.06
V_{G2}(p.u.)	1.04	1.03	1.05	1.05
V_{G5}(p.u.)	1.01	1.01	1.04	1.04
V_{G8}(p.u.)	1.01	1.02	1.04	1.04
V_{G11}(p.u.)	1.05	1.03	1.04	1.03
V_{G13}(p.u.)	1.05	1.05	1.04	1.04
T_{11}(p.u.)	1.08	0.94	1.01	1.05
T_{12}(p.u.)	1.07	1.04	0.99	0.91
T_{15}(p.u.)	1.03	0.96	0.98	0.98
T_{36}(p.u.)	1.07	0.96	0.99	0.99
Q_{c10} (MVAr)	0	4.99	2.98	0.02
Q_{c12}(MVAr)	0	3.18	2.87	4.93
Q_{c15}(MVAr)	0	1.43	2.07	2.12
Q_{c17}(MVAr)	0	5.00	3.24	3.38
Q_{c20}(MVAr)	0	5.00	1.91	2.06
Q_{c21}(MVAr)	0	5.00	4.23	4.99
Q_{c23}(MVAr)	0	1.98	0.88	0.82
Q_{c24}(MVAr)	0	0.04	3.95	3.85
Q_{c29}(MVAr)	0	1.27	0.48	0.97
Fcost($/h)	581.51	351.30	351.08	351.06
Ploss (MW)	1.69	1.80	1.72	1.71
TOC (s)	-	18.78	0.04	34.04

Table 5. OPF results for *Fcostvpe*-fuel cost minimization considering the valve-point effect in the IEEE 30-bus test system for load level of 1 (p.u.)

	Initial	PSOGSA	NNF	GA-*fmincon*
P_{G1}(MW)	99.22	219.81	219.77	219.82
P_{G2}(MW)	80	27.50	27.54	27.60
P_{G5}(MW)	50	15.86	15.89	15.74
P_{G8}(MW)	20	10	9.97	10.01
P_{G11}(MW)	20	10	10.00	10.01
P_{G13}(MW)	20	12	12.00	12.00
V_{G1}(p.u.)	1.05	1.09	1.09	1.09
V_{G2}(p.u.)	1.04	1.07	1.06	1.07
V_{G5}(p.u.)	1.01	1.03	1.03	1.03
V_{G8}(p.u.)	1.01	1.03	1.03	1.03
V_{G11}(p.u.)	1.05	1.08	1.08	1.09
V_{G13}(p.u.)	1.05	1.04	1.04	1.04
T_{11}(p.u.)	1.08	1.09	1.02	1.01
T_{12}(p.u.)	1.07	0.90	0.97	0.97
T_{15}(p.u.)	1.03	0.95	0.97	0.96
T_{36}(p.u.)	1.07	0.98	0.98	0.97
Q_{c10}(MVAr)	0	4.87	2.81	3.13
Q_{c12}(MVAr)	0	2.79	2.37	2.53
Q_{c15}(MVAr)	0	3.41	3.71	3.78
Q_{c17}(MVAr)	0	5	4.12	4.33
Q_{c20}(MVAr)	0	4.16	4.13	3.89
Q_{c21}(MVAr)	0	5	4.60	4.70
Q_{c23}(MVAr)	0	4.94	3.55	3.47
Q_{c24}(MVAr)	0	5	4.79	4.76
Q_{c29}(MVAr)	0	2.66	2.38	2.60
Fcostvpe ($/h)	960.19	824.15	824.22	824.16
Ploss (MW)	5.82	11.77	11.78	11.77
TOC (s)	-	31.04	0.03	39.23

Table 6. OPF results for *Fcostvpe*-fuel cost minimization considering the valve-point effect in the IEEE 30-bus test system for load level of 0.75 (p.u.)

	Initial	PSOGSA	NNF	GA-*fmincon*
P_{G1}(MW)	24.93	134.91	134.12	134.91
P_{G2}(MW)	80	34.64	35.05	34.28
P_{G5}(MW)	50	16.22	16.66	16.57
P_{G8}(MW)	20	10	9.91	10.00
P_{G11}(MW)	20	10	10.00	10.00
P_{G13}(MW)	20	12	12.00	12.00
V_{G1}(p.u.)	1.05	1.07	1.07	1.07
V_{G2}(p.u.)	1.04	1.06	1.06	1.06
V_{G5}(p.u.)	1.01	1.04	1.03	1.04
V_{G8}(p.u.)	1.01	1.04	1.04	1.04
V_{G11}(p.u.)	1.05	1.05	1.06	1.06
V_{G13}(p.u.)	1.05	1.04	1.04	1.04
T_{11}(p.u.)	1.08	1.063	1.02	0.99
T_{12}(p.u.)	1.07	0.92	0.99	0.99
T_{15}(p.u.)	1.03	0.97	0.99	0.97
T_{36}(p.u.)	1.07	0.98	0.99	0.98
Q_{c10}(MVAr)	0	4.98	2.59	4.31
Q_{c12}(MVAr)	0	0.01	2.69	4.15
Q_{c15}(MVAr)	0	4.39	3.08	3.04
Q_{c17}(MVAr)	0	5	3.57	4.23
Q_{c20}(MVAr)	0	3.21	2.82	3.23
Q_{c21}(MVAr)	0	5	3.69	4.85
Q_{c23}(MVAr)	0	1.55	2.37	2.06
Q_{c24}(MVAr)	0	5	4.34	4.82
Q_{c29}(MVAr)	0	1.65	1.45	1.71
Fcostvpe ($/h)	773.99	566.95	567.88	566.94
Ploss (MW)	2.38	5.23	5.12	5.211
TOC (s)	-	29.54	0.06	32.79

5 Conclusion

In this paper an NNF-based approach has been proposed and applied to fast evaluate the OPF solution in power system. First, a comparison of different metaheuristic methods was performed. The best method, namely PSOGSA, was selected to train the NNF network to estimate the OPF solution, using a large number of randomly selected load levels in the system. The effectiveness and computational efficiency of the proposed NNF-based OPF approach are demonstrated through case studies conducted on the IEEE 30-bus test system. Simulation results obtained on the standard IEEE 30-bus test system show that the application of the proposed NNF technique yields the same accuracy OPF results as the application of the classical metaheuristic methods PSOGSA or GA with the hybrid *fmincon* interior-point function in MATLAB® whose, but for a drastically shorter time. For example, for the IEEE 30-bus test system and load level of 1 p.u. the optimal costs obtained by applying PSOSGA and the proposed NNF approach are approx. 800.52 $/h. The OPF execution time using PSOGSA is 19.741 s, while using the NNF approach the execution time is much shorter, 0.03 s. The voltage limitations are satisfied, i.e., the voltages at all load nodes are within the permissible limits of (0.95 p.u., 1.05 p.u.) for both methods.

Proposed NNF-based approach enables a fast and efficient OPF evaluation suitable for real-time application in power systems. The advantage of this approach compared to training the NNF network using classical methods is that it is possible to include different types of variables, objective functions and constraints without the need for simplification and linearization of the mathematical model of OPF.

Acknowledgment

The author thanks the Ministry of Science, Technological Development and Innovation of the Republic of Serbia for funding the research (Contract No. 451-03-34/2026-03/200155) conducted by the Faculty of Technical Sciences, University of Priština.

References

[1] J. Radosavljević, (2024) Metaheuristic Optimization in Power Engineering, 2nd Edition *Volume* 1: Algorithms and power dispatch using MATLAB®-based software, The Institution of Engineering and Technology (IET), London, 2024.

[2] S.M. Miraftabzadeh, A.D. Martino, M. Longo and D. Zaninelli, "Deep learning in power systems: A bibliometric analysis and future trends", IEEE Access, vol 12, pp. 163172-163196, 2024.

[3] R. Nellikkath and S. Chatzivasileiadis, "Physics-informed neural networks for AC optimal power flow", Electric Power Systems Research, vol. 212, 108412, 2022.

[4] F. Cengil, H. Nagarajan, R. Bent, S. Eksioglu and B. Eksioglu, "Learning to accelerate globally optimal solutions to the AC optimal power flow problem", Electric Power Systems Research, vol. 212, 108275, 2022.

[5] A. Lotfi and M. Pirnia, "Constraint-guided deep neural network for solving optimal power flow", Electric Power Systems Research, vol. 211, 108353, 2022.

[6] X. Pan, T. Zhao, M. Chen and S.H. Low, "DeepOPF: A Deep Neural Network Approach for Security-Constrained DC Optimal Power Flow", IEEE Transactions on Power Systems, vol.36, no 3, pp. 1725–1735, 2021.

[7] W. Huang, X. Pan, M. Chen and S.H. Low, "DeepOPF-V: Solving AC-OPF Problems Efficiently", IEEE Transactions on Power Systems, vol. 37, no. 1, pp. 800-803, 2022.

[8] X. Pan, H. Wan, C. Minghua and H. Steven, "DeepOPF-AL: Augmented Learning for Solving AC-OPF Problems with a Multi-Valued Load-Solution Mapping", In Proceedings of the 14th ACM International Conference on Future Energy Systems, 42–47, New York, NY: ACM, 2023.

[9] X. Pan, M. Chen, T. Zhao and S.H. Low, "DeepOPF: A feasibility-optimized deep neural network approach for AC optimal power flow problems", IEEE Systems Journal, vol. 17, no. 1, no. 673-6833, 2023.

[10] Y. Zhou, B. Zhang, C. Xu, T. Lan, R. Diao, D. Shi, Z. Wang and W.J. Lee "A data-driven method for fast AC optimal power flow solutions via deep reinforcement learning", Journal of Modern Power Systems and Clean Energy, vol. 8, no 6, pp. 1128-1139, 2020.

[11] Y. Zhou, W.J. Lee, R. Diao and D. Shi, "Deep reinforcement learning based real-time AC optimal power flow considering uncertainties", Journal of Modern Power Systems and Clean Energy vol. 10, no. 5, pp. 1098-1109, 2022.

[12] Q. Tran, J. Mitra and N. Nguyen, "Learning model combining convolutional deep neural network with a self-attention mechanism for AC optimal power flow", Electric Power Systems Research, vol. 231,110327, 2024.

[13] Y. Chen, Q. Du, H. Liu, L. Cheng and M.S. Younis, "Improved proximal policy optimization algorithm for sequential security-constrained optimal power flow based on expert knowledge and safety layer", Journal of Modern Power Systems and Clean Energy vol. 12, no. 3, pp. 742-753, 2024.

[14] O.S.T. Al Butti, M. Burunkaya, J. Rahebi and J.M. Lopez-Guede, "Optimal power flow using PSO algorithms based on artificial neural networks", IEEE Access vol. 12, pp. 154778-154795, 2024.

[15] D. Owerko, F. Gama and A. Ribeiro, "Optimal power flow using graph neural networks", In: ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 04-08 May 2020, Barcelona, Spain.

[16] T.B. Lopez-Garcia and J.A. Domingez-Navarro, "Optimal power flow with physics-informed typed graph neural networks", IEEE Transactions on Power Systems, vol. 40, no. 1, pp. 381-393, 2025.

[17] G. Sun, Q. Wang, S. Chen, Z. Wei and H. Zang, "Look-ahead dispatch of power systems based on linear alternating current optimal power flow framework with nonlinear frequency constraints using physics-informed neural networks", Journal of Modern Power Systems and Clean Energy, vol. 13, no. 3, pp. 778-790, 2025.

[18] Y. Ye, Y. Wu, J. Hu, H. Hu, S. Qian, X. Zhang, Q. Wang and G. Strbac, "Physics-guided safe policy learning with enhanced perception for real-time dynamic security constrained optimal power flow", Journal of Modern Power Systems and Clean Energy, vol. 13, no. 5, no. 1507-1519, 2025.

[19] A. Tudoras-Miravet, E. Gonzalez-Iakl, O. Gomis-Bellmunt and E. Prieto-Araujo, “Physics-informed neural networks for power systems warm-start optimization”, IEEE Access, vol. 12, pp. 135913-135928, 2024.

[20] F.S. Al-Ismail “Physics informed neural networks-based AC optimal power flow under high RES penetration”, IEEE Access, vol. 12, pp. 189297-189306, 2024.

[21] T. Wu, A. Scaglione and D. Arnold, “Constrained reinforcement learning for predictive control in real-time stochastic dynamic optimal power flow”, IEEE Transactions on Power Systems, vol. 39, no. 3, pp. 5077-5090, 2024.

[22] K. Chen, S. Bose and Y. Zhang, “Physics-informed gradient estimation for accelerating deep learning-based AC-OPF”, IEEE Transactions on Industrial Informatics, vol. 21, no. 6, pp. 4649-4660, 2025.

[23] F. Fioretto, T.W.K. Mak and P. van Hentenryck, “Predicting AC optimal power flows: combining deep learning and lagrangian dual methods”, In: AAAI Conference on Artificial Intelligence 2020 - New York, USA, 7 Feb 2020-12 Feb 2020

[24] A. Velloso and P. Van Hentenryck, “Combining Deep Learning and Optimization for Preventive Security-Constrained DC Optimal Power Flow”, IEEE Transactions on Power Systems, vol. 36, no. 4, pp. 3618-3628, 2021.

[25] J. Radosavljević, D. Klimenta, M. Jevtić and N. Arsić, “Optimal power flow using a hybrid optimization algorithm of particle swarm optimization and gravitational search algorithm”, Electric Power Components and Systems, vol. 43, no. 17, pp. 1958-1970, 2015.

[26] L.L. Lai, J.T. Ma, R. Yokoyama and M. Zhao, “Improved genetic algorithms for optimal power flow under both normal and contingent operation states”. International Journal of Electrical Power and Energy Systems, vol. 19, no. 5, pp. 287-292, 1997.

Received 11 August 2026