

Real-Time Voltage Stability Monitoring Using Machine Learning-Based PMU Measurements



Mohammed Amroune, Arif Bourzami, Mohamed Zellagui,
and Ismail Musirin

Abstract Recently, due to the increasing demand with scarcity in installed production capacities, power systems are being operated closer to voltage stability limits resulting in a higher eventuality of voltage collapse. Thus, fast and accurate monitoring of voltage stability has become an important factor in the efficient operation of modern power systems. In this chapter, two approaches based on the combination of multi-layer perceptron (MLP) neural network and adaptive neuro-fuzzy inference system (ANFIS) with moth swarm algorithm (MSA) have been proposed to monitor voltage stability of power systems using phasor measurement units (PMUs) data. In the proposed hybrid MLP–MSA and ANFIS–MSA models, the MSA algorithm is adopted to optimize the connection weights and biases of the MLP network and to determine the tuning parameter in ANFIS model. To evaluate the prediction capability and efficiency of the proposed models, several statistical indicators such as root mean square error (RMSE), correlation coefficient (R) and root mean square percentage error (RMSPE) are used. Numerical studies are carried out on two standard power systems. The obtained results indicate that the proposed ANFIS–MSA model has the most reliable and accurate prediction ability and deemed to be the effective method to estimate the voltage stability margin of the power system based on measurements from PMU devices.

M. Amroune (✉) · A. Bourzami

Department of Electrical Engineering, University of Ferhat Abbas, Setif 1, 19000 Setif, Algeria
e-mail: mohammed.amroune@yahoo.fr

M. Zellagui

Département de Génie Électrique, École de Technologie Supérieure, Université de Québec,
Montréal, Canada

Department of Electrical Engineering, Faculty of Technology, University of Batna 2, Fesdis
05078, Batna, Algeria

I. Musirin

Faculty of Electrical Engineering, Universiti Teknologi MARA, 40450 Shah Alam, Selangor,
Malaysia

© The Editor(s) (if applicable) and The Author(s), under exclusive license
to Springer Nature Switzerland AG 2021

H. Haes Alhelou et al. (eds.), *Wide Area Power Systems Stability, Protection, and Security*,
Power Systems, https://doi.org/10.1007/978-3-030-54275-7_16

1 Introduction

The rapid growth of electricity demand, the environmental restrictions and the deregulation policies coerce the power systems to operate close to their stability limits. In such conditions, any contingency such as the loss of a generator or transmission line in the system may cause voltage collapse. Therefore, it is necessary to continuously monitor the voltage stability of the power system to avoid the risk of large blackouts. To monitor the voltage stability of the power system in real-time, the process of measurements collection and voltage stability margin (VSM) computation must be accomplished within the required time frame. Traditionally, supervisory control and data acquisition (SCADA) system have been used to collect measurements regularly every few minutes [1]. Therefore, real-time voltage stability monitoring is impractical with using traditional SCADA system. In recent years, the wide-area measurement system (WAMS) is increasingly being deployed in modern power systems (smart grids). With the prevalence of WAMS based on phasor measurement units (PMUs), the power system stability issues can be treated more efficiently. PMUs overcome the disadvantages of SCADA by providing the synchronized measurements of voltage and current phasors and frequency at a very high speed. The synchrophasor measurements gathered from PMUs can support the tracking of fast event and provide sufficient information for voltage stability monitoring. Voltage stability margin (VSM) estimation is one of the commonly used techniques for real-time voltage stability monitoring based on the provided PMU measurements. Many studies have been developed based on machine learning techniques to estimate the VSM in real-time. In [2], multi-layered perceptron (MLP) network based-back-propagation algorithm is introduced to estimate the VSM using the energy method. Joya et al. [3] utilized a sequential learning strategy to design a single MLP network to estimate the line voltage stability index for different load conditions. Venkatesan and Jolad [4] proposed the application of an MLP based model for fast voltage contingency ranking. The load flow equations are adopted, in this work, to determine the minimum singular values and the findings of the load flow analysis are used to train the MLP network. Authors in [5] proposed a novel MLP-based algorithm that involves a reduced number of inputs to estimate the voltage magnitude of weakest buses in the system. In [6], an effective technique based on Gram–Schmidt orthogonalization is proposed to find the optimal number of MLP inputs required to assure a good assessment of voltage stability. Adaptively trained MLP network is used in [7] as a mapping tool to approximate the available loading margin of the system. In this work, Z-score technique is applied to find and process any bad variable in the training dataset for the MLP network. Generally, ANN is considered as a powerful method for performing nonlinear regression. However, ANNs suffer from some drawbacks such as the amount of training time, the functional relationship which gets changed from one topology to another and the requirement of the appropriate values of weights and bias parameters [8, 9].

In the last years, many studies have been reported in the literature, exploiting the ability of support vector machine (SVM) technique for voltage stability monitoring.

Reference [10] discuss the evaluation of voltage stability using the regression version of SVM or the so-called support vector regression (SVR). Suganyadevi and Babulal [11] proposed the use of ν and ϵ types of SVR model with various kernel functions to estimate the VSM. In [12], least squares SVM (LS-SVM) with a reduced set of inputs is adopted to estimate the power system load-ability margin. Sajan et al. [9] developed a hybrid model integrating SVR with genetic algorithm (GA) for voltage stability evaluation. In the same way, we proposed in our previous works [13, 14] two hybrid models combining SVR with ant lion optimization (ALO) and dragonfly optimization (DFO) algorithms for voltage stability assessment. In these works, the ALO and DFO algorithms are adopted to find the appropriate SVR parameters. It was stated that the developed GA-SVR, ALO-SVR and DFO-SVR models have better performance compared to the MLP network. Although that SVM is a powerful and promising classification and regression tool, it suffers from overrun time and necessitates more memory for a big training dataset. On the other hand, the efficiency of the SVM model is highly depending upon the selected internal parameters [15].

Adaptive neuro-fuzzy inference system (ANFIS) is another powerful and flexible method proposed by some researchers for voltage stability monitoring. Modi et al. [16, 17] proposed the application of the ANFIS model to monitor the voltage stability of power systems incorporating FACTS devices. In [18], a fuzzy inference model is established and optimized by ANN and GA algorithm to assess the power system security margins. Authors of [19] adopted the subtractive clustering (SC) technique and ANFIS model to evaluate the VSM of the power system. Amroune et al. [14] introduce a method of utilizing ANFIS model-based synchrophasor measurements for on-line prediction of VSM. Even though with the good performance of the ANFIS model, its application in voltage stability analysis is still limited. The high computational costs and the complex set of its parameters are the major drawbacks of this method [11]. Therefore, the application of efficient methods to adjust ANFIS parameters will be of great importance since the unsuitable selection of these parameters can lead to inaccurate classification/regression.

The main purpose of this chapter is to present two real-time voltage stability monitoring approaches for secure and reliable power system operation. In the first approach, an improved multi-layer perceptron (MLP) neural network based on PMUs measurements is proposed to estimate the VSM of the power system in a real-time manner. In the proposed model, the moth swarm algorithm (MSA) [20] is integrated with the MLP network to optimize the connection weights and biases of the network to improve its performance. In the second approach, a novel hybrid model combining the adaptive neuro-fuzzy inference system (ANFIS) and MSA is proposed to monitor the voltage stability of power system. In the proposed hybrid model, the MSA algorithm is adopted to obtain proper parameter settings for the ANFIS-based subtractive clustering (SC) technique.

The rest of the chapter is organized as follows: Sect. 2 explains the standard structure of the MLP neural network, ANFIS and MSA algorithm. Section 3 describes the proposed hybrid MLP-MSA and ANFIS-MSA models. Section 4 describes the implementation of the proposed hybrid models for voltage stability monitoring. In

Sect. 5, the proposed hybrid models are validated and compared in IEEE 30-bus and IEEE 118-bus standard test systems. Finally a conclusion is drawn in Sect. 6.

2 Methods

This section presents the basic information of artificial neural network (ANN), adaptive neuro-fuzzy inference system (ANFIS) and moth swarm algorithm (MSA).

2.1 Artificial Neural Network (ANN)

2.1.1 Model of Neuron

Artificial neural network (ANN) is a type of machine learning techniques that simulates the mechanism of information management of the human brain system. A diagram of a neuron model with a single n -element input vector, which forms the modern basis for ANNs, is illustrated in Fig. 1. The main elements of the neural model are listed below:

- An input vector connected to a summation node via connecting links. Each of these links has an associated weight (w_i);
- A summation node in which the weighted input $w_i x_i$ is added to the scalar bias b to form the network input;
- An activation function (threshold) for limiting the amplitude of the output of the neuron.

The main objective of an activation function is to confirm that the neuron's response is bounded or limited. The activation functions are generally divided into

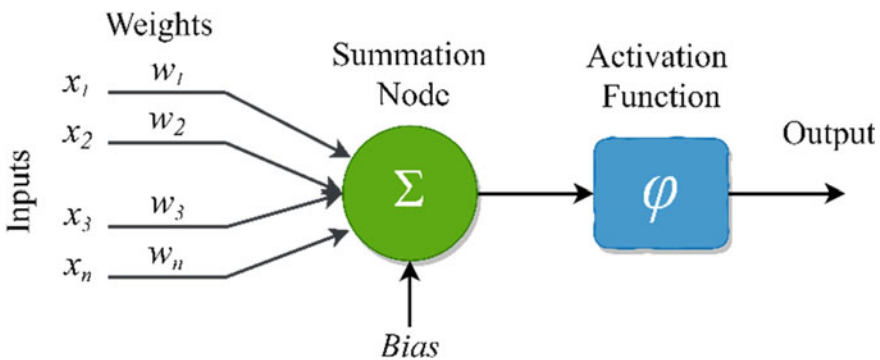


Fig. 1 Model of neuron

two main types linear and non-linear activation functions. Nonlinear functions such as logarithmic sigmoid, hyperbolic tangent sigmoid functions and pure linear function are the frequently utilized functions. The output (y) of the neuron governed by the activation function (φ) can be expressed as follows:

$$y = \varphi \left(\sum_{i=1}^n w_i x_i + b \right) \tag{1}$$

2.1.2 Multilayer Perceptron (MLP) Neural Network

One of the most used neural networks in engineering applications is the multilayer perceptron (MLP) neural network [21]. The structure of MLP includes one input layer, one output layer, and one or more hidden layers. The connections between the neurons are performed through some pre-specified weights. Figure 2 shows the typical planning of neurons in an MLP neural network. In this Figure, every node represents an artificial neuron. The neurons are organized in layers, there are one input layer, one output layer and multiple hidden layers. The relationship between the layers can be given by the following equations [22]:

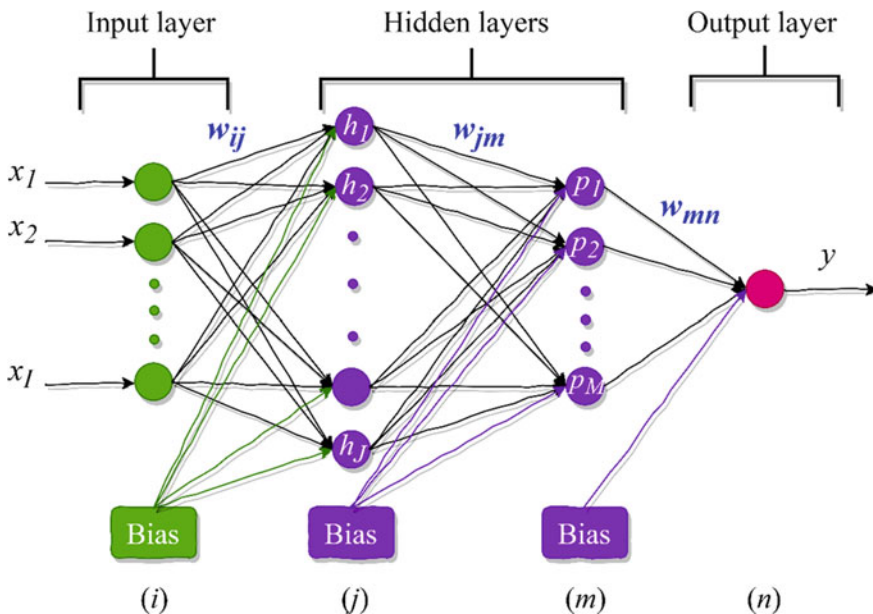


Fig. 2 Multilayer perceptron neural network

$$h_j = \varphi_{h1} \left(\sum_{i=1}^I w_{ij} x_i + b_{1j} \right), j = 1, \dots, K \quad (2)$$

$$p_m = \varphi_{h2} \left(\sum_{j=1}^K w_{jm} h_j + b_{2m} \right), m = 1, \dots, M \quad (3)$$

$$y_n = \varphi_{out} \left(\sum_{m=1}^M w_{mn} p_m + b_{3n} \right), n = 1, \dots, N \quad (4)$$

where w_{ij} , w_{jm} and w_{mn} are the associated weights, b_{1j} , b_{2m} and b_{3n} are the biases, $\varphi(x)$ denotes the activation function.

2.1.3 Training of MLP

Artificial neural networks are trained based on the relevant data by learning algorithms. During the training process, the weight and bias parameters are optimized. Then, these parameters are employed to process test dataset to obtain the final output. The MLP network learning can be divided into two main groups: supervised and unsupervised learning.

- Supervised learning—In this group, the system is presented with a set of inputs and the correct outputs, an external trainer controls the learning to learn a general rule that maps inputs to outputs. The weights are adjusted to minimize the error between the network outputs and the desired outputs;
- Unsupervised learning—In this group, there is no trainer involved and no labelled responses are given to the learning algorithm. Here the network is just exposed to a set of inputs and algorithms are left to their own to draw inferences.

In the supervised learning method and the one we use in this chapter, the weights and bias are adjusted to minimize the error between the actual and the predicted values in the next iteration. This process is repeated several times until the minimum error is achieved. Finally, the obtained weights and bias are utilized to carry out any tasks of the ANN i.e. classification or regression. There are several techniques to find the optimal values of the weights and bias by supervised learning. One of the vastly applied learning algorithms to train the MLP networks is the back-propagation (BP) algorithm. This method is based on the minimization of the error between the predicted and actual outputs by adjusting the weights. Notwithstanding its wide utilization, BP algorithm has some drawbacks such as the slow error convergence rate and the local minimum trap [23]. Therefore, there is a need for more robust and efficient optimization algorithms for MLP network training.

2.2 Adaptive Neuro-Fuzzy Inference System (ANFIS)

2.2.1 Overview of ANFIS

ANFIS was introduced by Jang in 1993 [24]. It is a machine learning technique incorporates the advantages of ANNs and fuzzy system. The fuzzy part generates a relationship between inputs and outputs, and the parameters associated with the membership part are specified by the neural network. Thence, the main features of both fuzzy and ANN methods are combined in this system.

The sample design of the ANFIS model with two inputs and two rules is shown in Fig. 3. It consists of five main layers; each layer contains several nodes designated by the node function. The functionality of these five layers is given as follows [24]:

Layer 1 (Fuzzification): In this layer, the inputs x and y are subjected to a membership function (e.g., triangle, trapezoidal, Gaussian). The generated output $O_{1,i}$, using generalized Gaussian membership function, can be expressed as follows:

$$O_{1,i} = \mu_{A_i}(x), i = 1, 2, O_{1,i} = \mu_{B_{i-2}}(y), i = 3, 4 \tag{5}$$

where μ_{A_i} and μ_{B_i} are Gaussian membership function given by:

$$\mu(x; c, \sigma) = e^{-\frac{1}{2}(\frac{x-c}{\sigma})^2} \tag{6}$$

where A_i and B_i are the membership values of the μ ; c and σ are the centre and width, respectively.

Layer 2 (Product): The output of each node in this layer is the product of all the received signals that are coming to this layer. This product can be computed using the following equation:

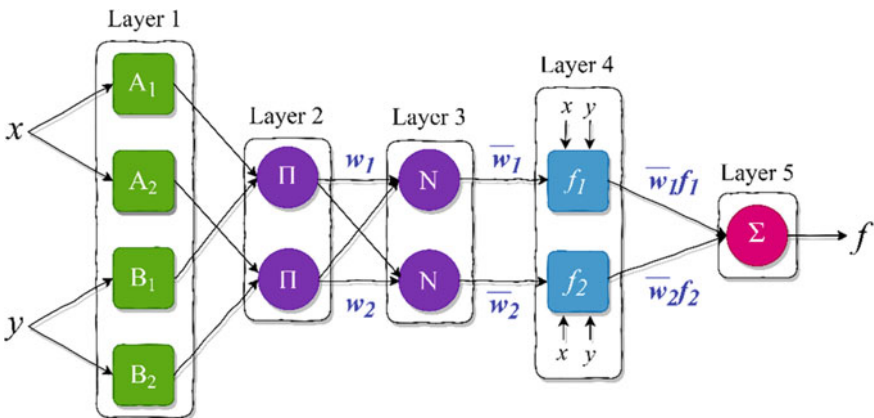


Fig. 3 The structure of ANFIS model for two inputs and two rules

$$O_{2,i} = \mu_{A_i}(x) \times \mu_{B_{i-2}}(y), \quad i = 1, 2 \quad (7)$$

Layer 3 (Normalization): In this layer, the output of the layer 2 is normalized using the following equation:

$$O_{3,i} = \bar{\omega}_i = \frac{\omega_i}{\sum_{i=1}^2 \omega_i}, \quad i = 1, 2 \quad (8)$$

Layer 4 (Defuzzification): The output of layer 3 is passed through the adaptive nodes of layer 4 as follows:

$$O_{4,i} = \bar{\omega}_i f_i = \bar{\omega}_i (p_i x + q_i y + r_i), \quad i = 1, 2 \quad (9)$$

where p , q and r are the consequent parameters of the i th node. These parameters are determined throughout the training phase.

Layer 5 (Overall output): Consists of a single node, which produce the overall output of the model.

$$O_5 = \sum_{i=1}^2 \bar{\omega}_i f_i = \frac{\sum_{i=1}^2 \bar{\omega}_i f_i}{\omega_1 + \omega_2} \quad (10)$$

2.2.2 Subtractive Clustering (SC)

The most crucial step in the developing of the ANFIS model is the generation of fuzzy inference system (FIS) with an optimum number and form of fuzzy rules to reduce the computational complexities. Thus, several methods such as grid partitioning, fuzzy c-means and subtractive clustering have been proposed to automate this process. Compared to the other algorithms, subtractive clustering (SC) [25] gives a better distribution of cluster centres and reduces the amount of data associated with the given problem. In this method, each data point is taken as a cluster centre candidate, afterwards, it computes the potential P_i of each data point x_i by determining the density of neighbouring points data using the following Equation.

$$P_i = \sum_{j=1}^m \exp\left(-\frac{\|x_i - x_j\|^2}{(r_a/2)^2}\right) \quad (11)$$

where m is the total number of data points in the N -Dimensional space. x_i and x_j are the data points, r_a is a positive constant defining a neighbourhood radius, and $\| \ \|$ represents the Euclidean distance. The data point with the highest potential value is

chosen as the first cluster centre x_{c1} and its density is P_{c1} . For the next cluster centre, the influence of the first cluster centre is subtracted to define the novel density values, as given by the Eq. 12:

$$P_i = P_i - P_{c1} \exp\left(-\frac{\|x_i - x_{c1}\|^2}{(r_b/2)^2}\right) \quad (12)$$

$$r_b = \eta \times r_a \quad (13)$$

where η is a positive number greater than 1.

According to Eq. 12 all the points close to the measured cluster centre x_{c1} will have low potential values and therefore they will not be taken as the next cluster centres. The next cluster centre x_{c2} is chosen after the recalculation of the potential of each data point. This process is repeated until sufficient cluster centres are produced.

2.3 Moth Swarm Algorithm (MSA)

Moth swarm algorithm (MSA) is a novel meta-heuristic optimization method proposed in 2017 by Ali Mohamed et al., [26] as a developed version of moth flame optimizer [20]. This algorithm is inspired by the navigational behaviour of moths in nature. The position of the light is expressed as the optimal solution, and the brilliance of this light is considered as the objective function. MSA algorithm comprises three collections of moths, which are defined as follows:

Pathfinders: A small group of moths that has the aptitude to find out the new areas over the optimization space and to discover the best position as the light source and to lead other individuals in the population to this position.

Prospectors: This second group is taking charge of wandering into arbitrary spiral paths set by the first group.

Onlookers: This group of the moths drift directly toward the best global solution which has been determined by prospectors.

Through the iterations, each moth is integrated into the optimization problem to search for the luminescence intensity of its corresponding light source. Pathfinders' positions are taken as the best fitness values, while the second and third best fitness take the names of prospectors and onlookers, respectively. The MSA is represented in four main phases [20]:

2.3.1 Initialization

The initial position of moths is selected randomly as:

$$X_{ij} = rand[1, 0] \times (X_j^{\max} - X_j^{\min}) + X_j^{\min} \forall i \in \{1, 2, \dots, n\}, j \in \{1, 2, \dots, d\} \tag{14}$$

where n is the number of populations and d is the dimension of the problem.

After initialization, the type of moth in the swarm is chosen based on the calculation of objective function. The best value of the objective function is selected to be pathfinders, and others are selected to be prospectors and onlookers.

2.3.2 Reconnaissance

In this phase, the pathfinders are updating their positions through the following five steps. In the first step, a proposed diversity index is employed to select the crossover points. The normalized dispersal degree at t iteration can be expressed as follows:

$$\sigma_j^t = \sqrt{\frac{\frac{1}{N_p} \sum_{i=1}^{N_p} (X_{ij}^t - \bar{X}_j^t)^2}{\bar{X}_j^t}} \tag{15}$$

The variation coefficient can be computed as follows:

$$\bar{X}_j^t = \frac{1}{N_p} \sum_{i=1}^{N_p} X_{ij}^{t2} \tag{16}$$

where N_p is the number of pathfinders:

$$\mu^t = \frac{1}{d} \sum_{j=1}^d \sigma_j^t \tag{17}$$

where μ^t is the variation degree of the relative dispersion.

In the second step, the random processes based on α -stable distribution are explained as Lévy flights [27].

The third step is called difference vectors Lévy mutation in which the sub-trial vectors are generated based on host vectors and donor vectors.

$$v_{pj}^t = \begin{cases} v_{pj}^t & \text{if } j \in c_p \\ x_{pj}^t & \text{if } j \notin c_p \end{cases} \tag{18}$$

In the fourth step, the position of each pathfinder is updated based on an adaptive crossover.

In the final step, a selection strategy is applied to define the best solutions to survive the next generation as follows:

$$\vec{x}_p^{t+1} = \begin{cases} \vec{x}_p^t & \text{if } f(\vec{v}_p^t) \geq f(\vec{x}_p^t) \\ \vec{v}_p^t & \text{if } f(\vec{v}_p^t) < f(\vec{x}_p^t) \end{cases} \quad (19)$$

The probability value P_p is estimated as follows:

$$P_p = \frac{fit_p}{\sum_{p=1}^{n_p} fit_p} \quad (20)$$

The luminescence intensity is computed from the fitness function of the problem f_p as follows:

$$fit_p = \begin{cases} \frac{1}{1 + f_p} & \text{for } f_p \geq 0 \\ 1 + |f_p| & \text{for } f_p < 0 \end{cases} \quad (21)$$

2.3.3 Transverse Orientation

In this phase, the moths with the minimal luminosity of light are taken as prospectors, and their numbers n_f reduced throughout iterations as follows:

$$n_f = \text{round}\left(\left(n - n_p\right) \times \left(1 - \frac{t}{T}\right)\right) \quad (22)$$

where n_p is the number of pathfinders, t is the current iteration, T is the number of iterations.

The position of each prospector is updated according to the spiral flight path as follows:

$$x_i^{t+1} = |x_i^t - x_p^t| \cdot e^\theta \cdot \cos 2\pi\theta + x_p^t \quad (23)$$

where $\theta \in [r, 1]$ is a random number to define the spiral shape and $r = -1 - t/T$.

2.3.4 Celestial Navigation

In this phase, the number of prospectors is decreased and the number of onlookers is increased. Moth with low fitness value is considered the onlooker and it can be computed using Eq. (24).

$$N_o = n - N_s - N_p \quad (24)$$

The onlooker contains the two following groups:

(1) The first group fly according to Gaussian distribution with $N_G = N_o/2$

$$f(q) = \frac{1}{\sqrt{2\pi uG}} \exp\left(-\frac{(qu)^2}{2\sigma_G^2}\right) - \infty < q < \infty$$

$$(q \sim N(\mu, \sigma_G^2)) \quad (25)$$

$$x_i^{t+1} = x_i^t + \varepsilon_1 + [\varepsilon_2 \times best - \varepsilon_3 \times x_i^t]$$

$$\forall i \in \{1, 2, \dots, N_G\} \quad (26)$$

$$\varepsilon_1 \sim random(size(d)) \oplus \left(best_g^t, \frac{\log t}{t} \times (x_i^t - best_g^t) \right) \quad (27)$$

where ε_1 is the random sample from Gaussian distribution, *best* is the global best solution (moonlight) which is obtained by transverse orientation and ε_2 and ε_3 are random numbers that range from [0, 1].

(2) The second group with size $N_A = N_o - N_G$

The updating equation for this group can be given as:

$$x_i^{t+1} = x_i^t + 0.001 \cdot G \left[x_i^t - x_i^{\min}, x_i^{\max} - x_i^t \right] + (1 - g/G) \cdot r_1$$

$$\cdot (best_p^t - x_i^t) + 2g/G \cdot r_2 \cdot (best_g^t - x_i^t) \quad (28)$$

where $i \in \{1, 2, \dots, N_A\}$.

$2g/G$ is the social factor, $1-g/G$ is the cognitive factor, r_1 and r_2 are randomly chosen numbers in the space [0, 1], $best_p$ is the arbitrarily chosen light source from the novel pathfinders group based on the probability value of its corresponding solution. At the end of every iteration, the type of each moth is redefined for the upcoming iteration.

3 Proposed Hybrid Models

3.1 MSA for Training MLP Network

As the MLP training is one of the main challenges in the use of this method, the appropriate values of weights and bias parameters must be defined to improve the efficiency of the MLP network [28]. One of the widely applied learning algorithms to find the optimum values of the weights and bias parameters is the back-propagation (BP) algorithm. However, the BP algorithm has some drawbacks such as the slow error convergence rate and the local minimum trap [28]. Several optimization methods have been proposed in the literature to enhance the performance of the neural networks, such as simulated annealing (SA), tabu search (TS), genetic algorithm (GA) and others. Therefore, the improvement of the performance of the MLP network can be achieved by replacing the conventional algorithms used in the training of MLP by more efficient optimization algorithms. In this section, a detailed description of the training process of the MLP network using MSA algorithm is presented. Two main phases are considered when the MSA algorithm is adopted to train MLP network, the first one is the representation of the search agents in the MSA and the last one is the choice of the fitness function. In MSA algorithm each search agent (moth) is encoded to represent the MLP candidate (weights and bias). Therefore, control vectors include a set of weights and a set of biases. The length of each vector is equal to the total number of weights and biases which depends on the number of input variables and the number of hidden layer neurons. The root mean square error (*RMSE*) is used as a fitness function. This assessment metric computes the difference between actual and predicted values by MLP-MSA model. *RMSE* is given by the following equation:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (a_i - p_i)^2} \quad (29)$$

where n designates the total number of data, a and p represent the actual and the predicted outputs, respectively.

The flowchart of the proposed MLP-MSA prediction model is shown in Fig. 4.

3.2 Hybrid ANFIS-MSA Model

Although ANFIS is a powerful mathematical tool for data regression and function estimation. Compared to other algorithms such as k-means clustering and fuzzy c-means, ANFIS-based SC gives a better distribution of the cluster centre and reduces the amount of data associated with the given problem. However, there is no standard rule to select its parameters, which is considered as the main blowbacks of this

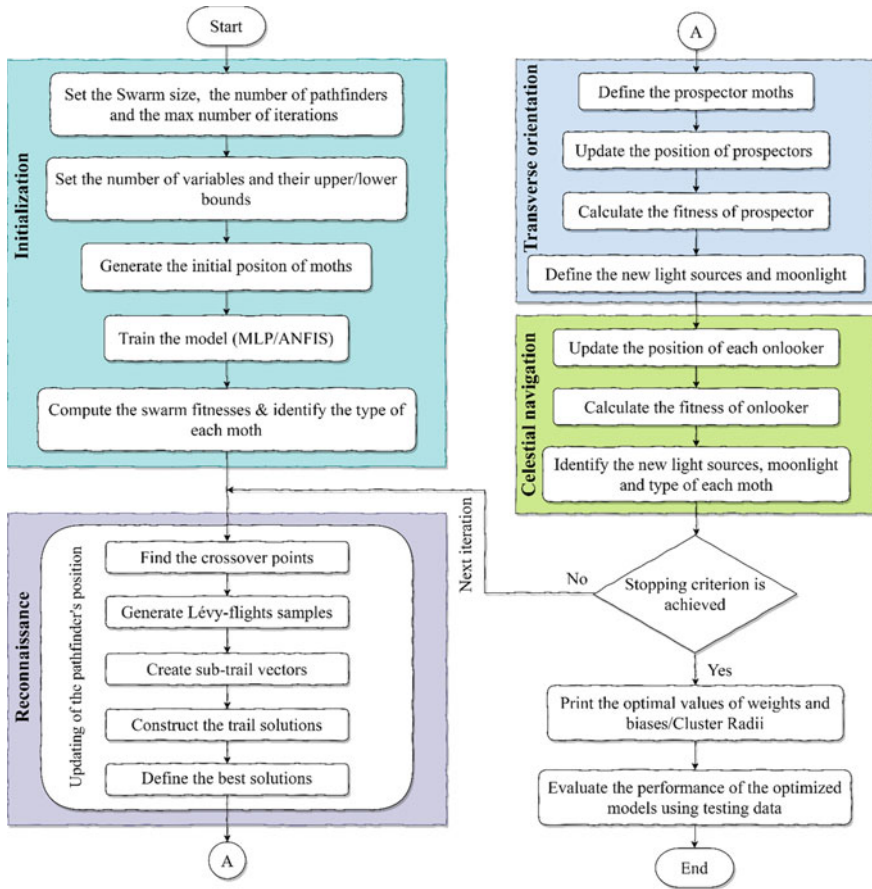


Fig. 4 Flowchart of the proposed hybrid models

method. Cluster radius parameter is one of these parameters that highly influenced on the complexity and generalization abilities of the ANFIS model. A small cluster radius results in small clusters in the data and, hence, many fuzzy rules. Large cluster radius yields few large clusters in the data which means fewer fuzzy rules [29]. Therefore, the application of an efficient method to adjust cluster radii will be of great importance. In this study, the MSA algorithm will be used to tune the cluster radii parameter of the ANFIS model. The proposed hybrid ANFIS–MSA model starts by generating the initial position of moths, includes the initial values of cluster radii, using Eq. (14). The next step involves the training of the ANFIS model, the calculation of the swarm fitness and the identification of the type of each moth. Before these steps, the data is divided into two sets of training and testing. The *RMSE*, expressed by Eq. (29), is used as a fitness function. The detailed flow diagram of the proposed model is given in Fig. 4.

3.3 Accuracy Assessment Criteria

To assess the accuracy of the proposed models, four performance criteria are used. These performance criteria are summarized as follows:

3.3.1 Root Mean Square Error (RMSE)

RMSE is the most commonly utilized measure of the differences between the values predicted by a model and the actual values. The model with the small value of *RMSE* is considered the best. The *RMSE* index is given by the Eq. (29).

3.3.2 Correlation Coefficient (R)

The correlation coefficient (*R*), with a value in the range [0, 1], delivers good information about the accuracy of the machine learning models. A value closer to 1 designates a good accuracy of the model. The *R* index is expressed as follows:

$$R = \frac{\sum_{i=1}^n (a_i - \bar{a})(p_i - \bar{p})}{\sqrt{\sum_{i=1}^n (a_i - \bar{a})^2 \sum_{i=1}^n (p_i - \bar{p})^2}} \quad (31)$$

where \bar{a} and \bar{p} are the rate of the actual and the estimated values, respectively.

3.3.3 Percent Root Mean Square Error (PRMSE)

PRMSE measures the accuracy of a machine learning method as a percentage, and it can be given by:

$$PRMSE = \frac{RMSE}{\sqrt{\frac{1}{n} \sum_{i=1}^n p_i^2}} \times 100 \quad (32)$$

The model accuracy is excellent for $PRMSE < 10\%$, good for $10\% < PRMSE < 20\%$, reasonable for $20\% < PRMSE < 30\%$ and low for $PRMSE > 30\%$.

4 Models Implementation for Voltage Stability Monitoring

4.1 Voltage Stability Indicator

In recent years, several indices have been developed to evaluate voltage stability status, to predict voltage stability margin and to identify the weak buses/area in the system. According to [30], the on-line voltage stability index (VSI) proposed by Yanfeng et al. [31], can be considered as one of the best line voltage stability indices. This index indicates the variation of voltage stability margin in the power system for real, reactive and apparent powers transmitted in the line. VSI is expressed as follows [31]:

$$VSI = \min\left(\frac{P_{\max} - P_r}{P_{\max}}, \frac{Q_{\max} - Q_r}{Q_{\max}}, \frac{S_{\max} - S_r}{S_{\max}}\right) \quad (33)$$

where

$$P_{\max} = \sqrt{\frac{V_s^4}{4X} - Q_r \frac{V_s^2}{X}} \quad (34)$$

$$Q_{\max} = \frac{V_s^2}{4X} - \frac{P_r^2 X}{V_s^2} \quad (35)$$

$$S_{\max} = \frac{(1 - \sin(\theta))V_s^2}{2 \cos(\theta)^2 X} \quad (36)$$

where P_{\max} , Q_{\max} , and S_{\max} are, respectively, the maximum transferred real, reactive and apparent powers, V_s and V_r are, respectively, the sending and the receiving end voltages, X is the line reactance, θ is the line impedance angle. The value of VSI index must be greater than 0 for stable systems and the branch with the lower value is considered to be weak compared to the branch with a higher value.

4.2 Generation of Training and Testing Data

As PMUs have been widely implemented in many power systems, application of wide-area PMU measurements in real-time voltage stability monitoring has been of great interests. Many studies confirmed that PMU data are good indicators for voltage stability monitoring and they can be taken as inputs to the prediction models [1, 13, 14]. Since PMU devices can provide synchronized measurements, which include the magnitude and phase angle of voltages, both of them are chosen as inputs of the developed MLP-MSA and ANFIS-MSA models i.e., inputs = $\{|V_i|, \delta_i\}$, $i \in$ PMU buses}. On the other hand, the minimum VSI values, computed using load flow

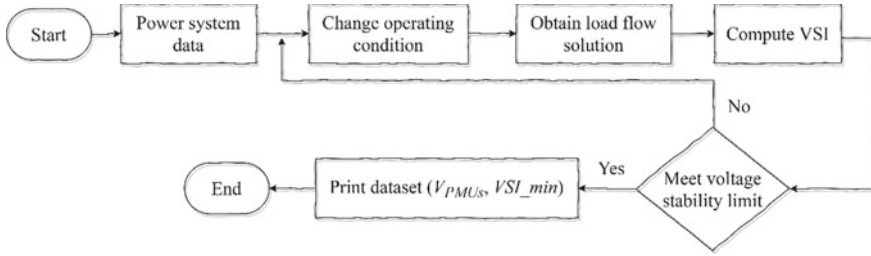


Fig. 5 Flowchart of the generation of training and testing data

equations, at each operating point are used as the output variables. The generation of training data is carried out through off-line simulation processes by varying both the active and reactive power simultaneously on each load bus in the system. The load is increased with a constant load factor from the base case until the system reaches the voltage stability limit. The voltage magnitudes and angles of PMU buses are obtained by solving conventional load flow at each load generating sample. The flowchart of the generation of training and testing data is depicted in Fig. 5. The collected dataset will be then applied to train and to evaluate the proposed prediction models. Once the training process is accomplished and the stopping condition is reached, model testing is required to verify the performance of the models over the actual and predicted data.

4.3 Real-Time Prediction of VSI

The last phase deals with the implementation of the developed MLP–MSA and ANFIS–MSA models to predict VSI in a real-time manner. In this phase, VSI is predicted using real-time measurements provided by PMUs. These provided data may support the tracking of dynamic phenomena and provide the necessary information for power system voltage stability monitoring. The time-synchronized data taken from throughout the distributed PMU units will be sent to the control system in which the well trained MLP and ANFIS models are employed to predict voltage stability margin for each operating point. The precise and synchronized real-time measurements obtained by PMUs with the fast evaluation of voltage stability offered by the proposed models can help system operators to take the required control action, such as load shedding or emergency demand response [32], to prevent voltage collapse.

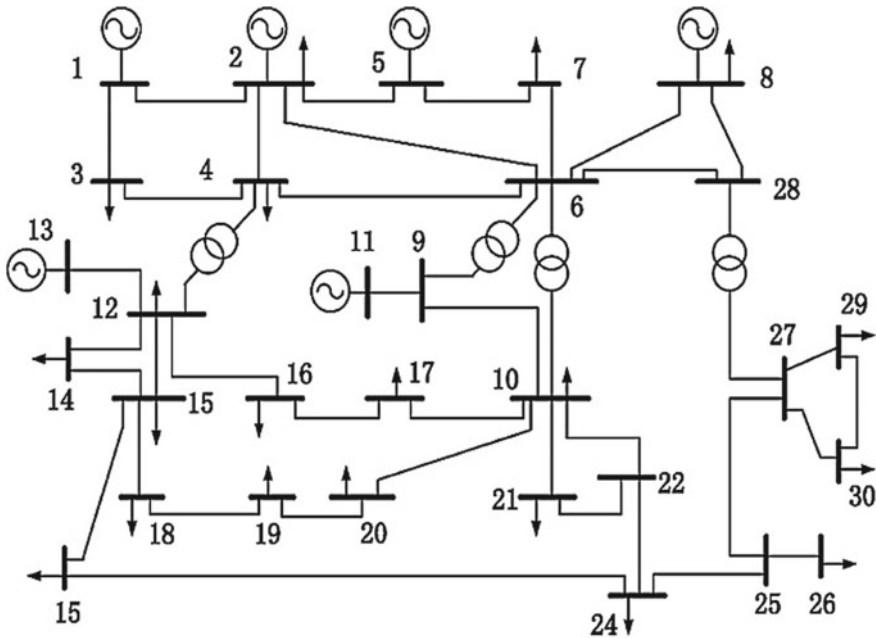


Fig. 6 Single line diagram of the IEEE 30-bus test system

5 Case Study and Simulation Results

5.1 Test Systems

5.1.1 IEEE 30-Bus Test System

The first test system used to validate the performance of the proposed models is the standard IEEE 30-bus test system [33]. This system is shown in Fig. 6, and it consists of 30 buses, 6 thermal units, 41 branches and 21 loads.

5.1.2 IEEE 118-Bus Test System

The performance of the proposed models for voltage stability monitoring has been validated also on the IEEE 118-bus test system shown in Fig. 7 [33]. This system contains 118 buses, 51 thermal units, 196 branches and 91 loads.

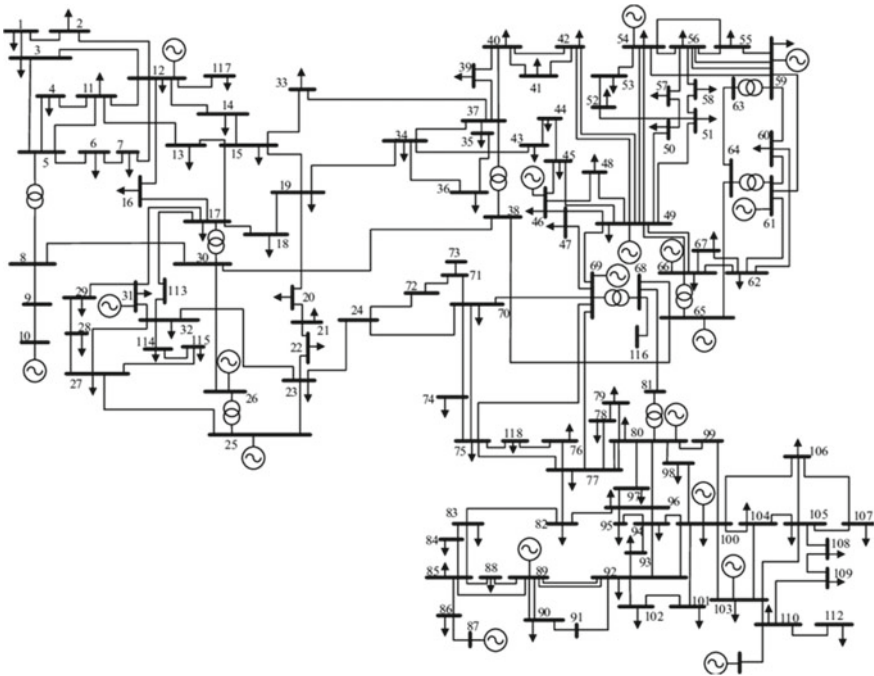


Fig. 7 Single line diagram of the IEEE 118-bus test system

5.2 Data Preparation

One of the crucial factors for successful implementation of any machine learning technique is the generation of proper training data. To generate the training data for the proposed models, all loads of both test systems are uniformly increased, with constant load power factors, from their base case loadings to the voltage collapse. As aforementioned, the gathered voltage magnitudes and angles by the distributed PMUs (voltages where PMUs are installed) will be used as the inputs for the proposed models, while the min values of VSI as the outputs. The optimal number and locations of PMUs for both test systems are obtained using simulated annealing (SA) method in PSAT (Power System Analysis Toolbox) software [34]. The optimal number and placement of PMUs are represented in Table 1. Afterwards, the collected data is divided into 80% for training and 20% for testing the models. The first portion is

Table 1 Number and locations of PMUs

Test system	Number of PMUs	Location of PMUs
IEEE 30-bus system	7	3, 5, 10, 12, 19, 23, 27
IEEE 118-bus system	28	2, 8, 11, 12, 17, 21, 25, 28, 33, 34, 40, 45, 49, 52, 56, 62, 72, 75, 77, 80, 85, 86, 90, 94, 101, 105, 110, 114

employed for initial training and models parameter optimization, and the second portion to further adjust and evolve the predictors to simulate real conditions.

5.3 Performance Comparison

The performance of MLP–MSA and ANFIS–MSA is outlined in this section. The developed programs are written using MATLAB software and the simulations are carried out on a computer with Intel Core i5 CPU @2.7 GHz, 4 GB RAM and Windows7 as the operating system. The load flow is obtained using MATPOWER [33]. In all simulations, the MSA parameters are tuned as follows: the number of search agents (candidate solutions) is fixed to 30 and the number of pathfinders to 18. For the MLP network, the number of hidden neurons is set to 20 (determined by a trial-and-error process). The MSA is used to optimize the weights connecting the input layer with the hidden layer, the weights connecting the hidden layer with the output layer and the biases. For the ANFIS model development, the SC (genfis2) technique based on the Gaussian type of membership function is used to generate fuzzy rules. According to [35] Gaussian membership function can be considered as the best fit to use with ANFIS model. The squash factor, the accept ratio and the reject ratio were set, by default in MATLAB toolbox, to 1.25, 0.5 and 0.15, respectively. The MSA algorithm is adopted to find the best value of cluster radii in the range of [0.2 0.5] [36].

5.3.1 Application to the IEEE 30-Bus Test System

This section demonstrates the effectiveness of the proposed methods on the IEEE 30-bus system. The results of the proposed methods have been compared with those found in the literature using the same data as in [14]. Figure 8a shows the convergence curve of MSA algorithm seeking for the optimal values for MLP’s weights and biases. It can be seen that the MSA has a slow convergence rate due to the problem

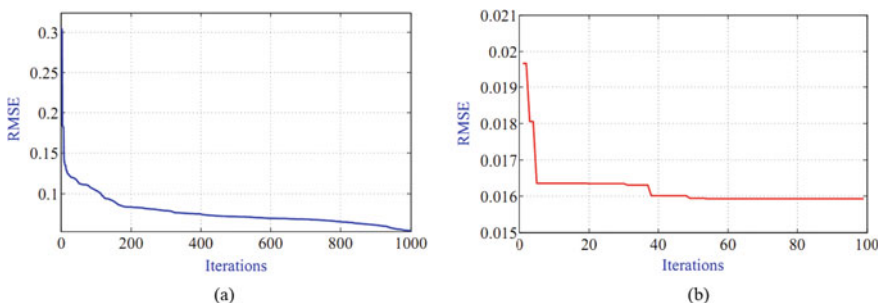


Fig. 8 Convergence curves of MSA for **a** MLP training, **b** ANFIS training

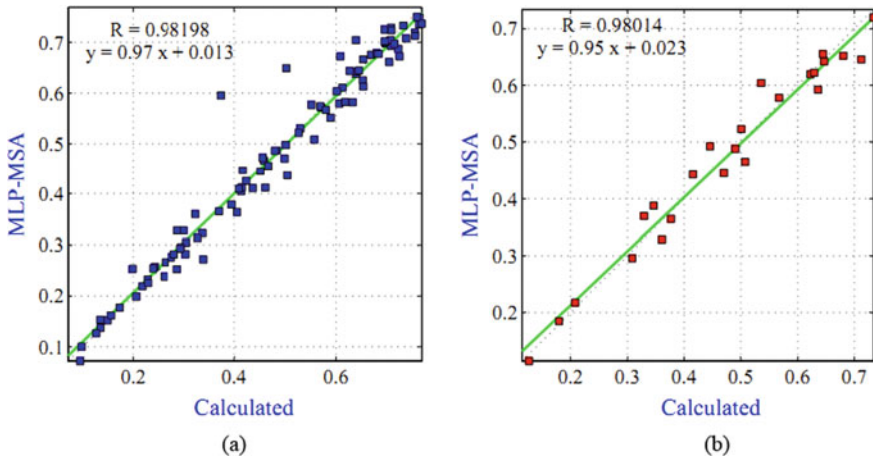


Fig. 9 Predicted values by the MLP-MSA versus actual values **a** Training phase **b** Testing phase

complexity. Figure 9a, b show the plots of the actual and predicted values of VSI via MLP-MSA method in the training and testing phases. It is seen that the MLP-MSA predictions are in agreement with the actual values. The prediction accuracy of the models was measured using *RMSE*, *PRMSE* and *R* indices. In the training phase, the computed values of *RMSE*, *PRMSE* and *R* were 0.0372, 6.1137 and 0.98198, respectively. In the testing phase, these indices are found to be 0.0380, 7.1458 and 0.98014 respectively. It is seen that the MLP-MSA model can estimate VSI with a good accuracy which means that the MSA algorithm has performed efficiently in tuning the MLP's weights and biases.

MSA algorithm is used also to find the optimal cluster radius of ANFIS-based SC technique, then, the performance of the trained ANFIS model was evaluated in VSI prediction. The convergence curve of the MSA algorithm is illustrated in Fig. 8b and the smallest error was obtained with the cluster radius of 0.2518. Figure 10a, b depicts the scatter plot of the predicted VSI values using ANFIS-MSA against actual ones in the training and testing phases. For the training step, the correlation coefficient (*R*) was found to be 0.99685. In the testing phase, the predictions result in a correlation coefficient of 0.99378. The correlation between the actual VSI values and ANFIS-MSA predictions is much better than MLP-MSA predictions. The comparison of different statistical indices for the proposed models and other models in the literature is shown in Table 2. As shown in this table, ANFIS-MSA outperforms all other prediction models. For this model, the values of *RMSE* and *PRMSE*, in the training phase, are calculated to be 0.0160 and 2.7873, respectively. In the testing phase, these values are found to be 0.0206 and 3.9442, respectively. It is seen that all used criteria confirmed that the proposed MLP-MSA and ANFIS-MSA methods give a good prediction accuracy compared to ANFIS and DFO-SVR [14]. On the other hand, the ANFIS-MSA performance culminates in giving the best prediction results than the MLP-MSA model.

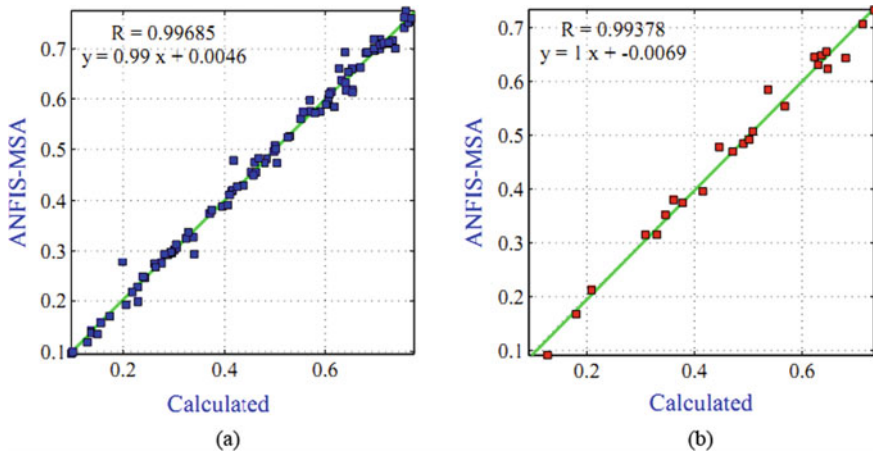


Fig. 10 Predicted values by the ANFIS–MSA versus actual values **a** Training phase **b** Testing phase

Table 2 Statistical performance of MLP–MSA and ANFIS–MSA

Method	Training			Testing		
	<i>RMSE</i>	<i>PRESE</i>	<i>R</i>	<i>RMSE</i>	<i>PRMSE</i>	<i>R</i>
MLP–MSA	0.0372	6.1137	0.98198	0.0380	7.1458	0.98014
ANFIS–MSA	0.0160	2.7873	0.99685	0.0206	3.9442	0.99378
ANFIS [14]	0.0228	–	0.99325	0.0385	–	0.97492
DFO–SVR [14]	0.0166	–	0.99641	0.0273	–	0.98776

5.3.2 Application to the IEEE 118-Bus Test System

The proposed new models are evaluated in this section on the IEEE 118-bus power system to compare their efficiency. Figure 11a illustrates the convergence curve of MSA algorithm searching for the optimal values for MLP’s weights and biases. Correlations between actual and predicted values of VSI for training and testing phases are shown in Fig. 12a, b. As seen from this Figure, the MLP–MSA model gives a correlation coefficient of 0.99814 for the training data while this coefficient for the testing data is equal to 0.94607. The proposed hybrid model managed to produce an *RMSE* and *PRMSE* of 0.0087411 and 1.5056, respectively, in the training phase. In the testing phase, this model yield to the values of *RMSE* and *PRMSE* of 0.1381 and 9.6441, respectively.

The effectiveness of the proposed ANFIS–MSA model in the prediction of VSI was also checked in the case of the IEEE 118-bus system. Figure 11b displays the optimization process of the *RMSE* versus iterations number for MSA to find the optimum value of ANFIS cluster radius and to predict VSI. The optimal found value of cluster radius is 0.4599. The scatter plots for this model are illustrated in Fig. 13a, b.

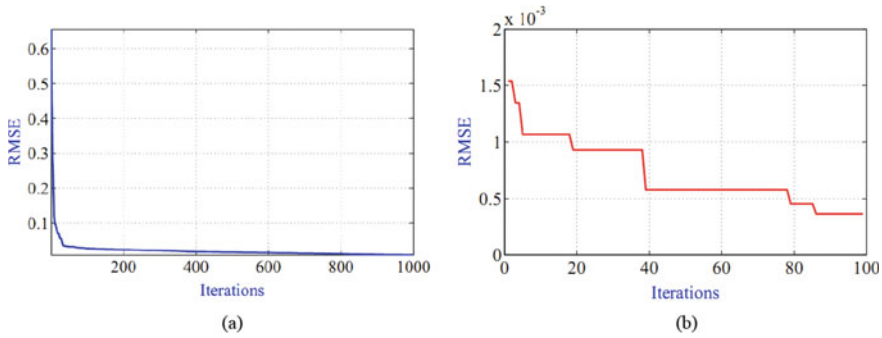


Fig. 11 Convergence curves of MSA for **a** MLP training, **b** ANFIS training

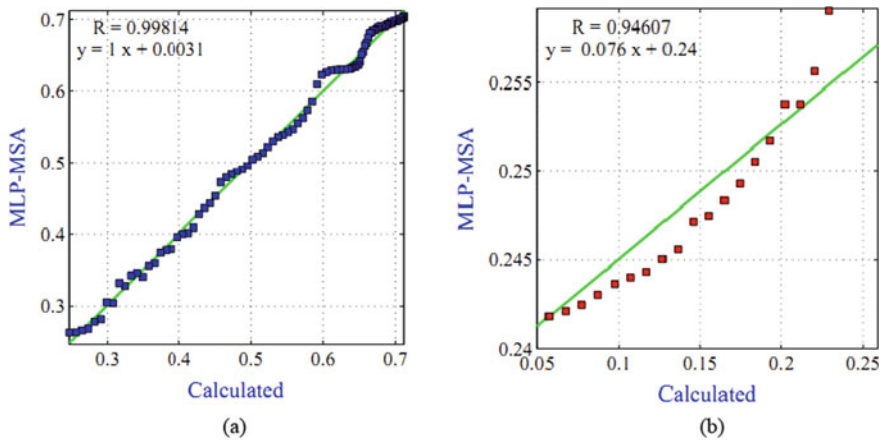


Fig. 12 Predicted values by the MLP-MSA versus actual values **a** Training phase **b** Testing phase

From this Figure, it is apparent that the ANFIS-MSA model gave the most scattered estimates. The results of different statistical indices for both proposed prediction models are listed in Table 3. It is shown that the *RMSE* and *PRMSE* of the ANFIS-MSA prediction method are computed to be $3.6708e^{-4}$ and 0.0632, in the training phase and 0.0042 and 2.8645 in the testing phase. By using ANFIS-MSA method, we observed that the *RMSE* in the testing phase was reduced by 0.1339. As for the *R*, it increased by 0.05392. From these results, we can note clearly that the prediction of VSI using ANFIS-MSA model gives better results compared to the MLP-MSA model. It is seen from the comparison of the statistical indicators values, in both case studies of IEEE 30-bus and IEEE 118-bus test systems, that the proposed models give a good prediction accuracy of VSI. However, ANFIS-MSA model acquired relatively lower values of *RMSE* and *PRMSE*, and higher values of *R*. Accordingly, it can draw the conclusion that the proposed MLP-MSA and ANFIS-MSA models

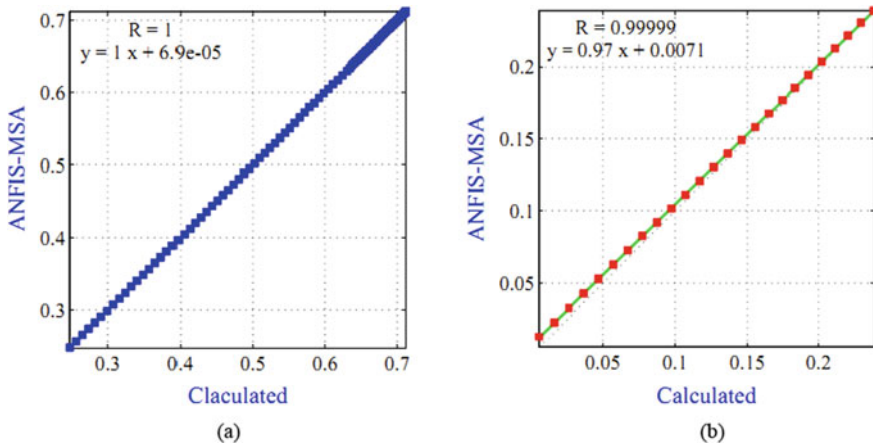


Fig. 13 Predicted values by the ANFIS–MSA versus actual values **a** Training phase **a** Testing phase

Table 3 Statistical results of the developed models

Method	Training			Testing		
	<i>RMSE</i>	<i>PRMSE</i>	<i>R</i>	<i>RMSE</i>	<i>PRMSE</i>	<i>R</i>
MLP–MSA	0.0087411	1.5056	0.99814	0.1381	9.6441	0.94607
ANFIS–MSA	$3.6708e^{-4}$	0.0632	1	0.0042	2.8645	0.99999

would be an appealing option for voltage stability monitoring since the obtained results are superior to those from the other models for the considered case studies.

6 Conclusions

This chapter proposes novel measurement-based methods for the real-time monitoring of voltage stability using PMU data. The proposed methods are based on the training of multi-layer perceptron (MLP) neural networks and adaptive neuro-fuzzy inference systems (ANFIS) deploying a moth swarm algorithm (MSA). The problem of training the MLP and ANFIS was first formulated as a minimization problem. The objective was to minimize the root mean square error (RMSE), and the parameters were linking weights and biases for the MLP network and the cluster radius for ANFIS-based subtractive clustering. The proposed MLP–MSA and ANFIS–MSA methods require only the voltage phasors’ data provided by the PMU units, to predict the voltage stability of the power system. The applicability and the performance of the proposed hybrid models have been investigated using standard IEEE 30-bus and IEEE 118-bus test systems and compared with existing methods in the literature.

The results obtained from the two test systems clearly showed the efficiency and accuracy of the proposed MLP–MSA and ANFIS–MSA methods as compared to other existing methods. On the other hand, the simulation results reveal that the ANFIS–MSA performance culminates in the best prediction results compared to the MLP–MSA model.

References

1. H. Su, T.Y. Liu, Enhanced-Online-Random-Forest Model for Static Voltage Stability Assessment Using Wide Area Measurements. *IEEE Transactions in Power Systems* **33**(6), 6696–6704 (2018)
2. B. Jeyasurya, Artificial neural networks for power system steady-state voltage instability evaluation. *Electr. Power Syst. Res.* **29**, 85–90 (1994)
3. G. Joya, F. García-Lagos, F. Sandoval, Contingency evaluation and monitorization using artificial neural networks. *Neural Comput. Appl.* **19**(1), 139–150 (2007)
4. M. Venkatesan, B. Jolad, Artificial neural network-based contingency ranking. *Comput Netw Inf Technol* **1**(142), 33–38 (2011)
5. P. Aravindhbabu, G. Balamurugan, ANN-based voltage estimation. *Appl. Soft. Comput.* **12**, 313–319 (2012)
6. A. Bahmanyar, A. Karami, Power system voltage stability monitoring using artificial neural networks with a reduced set of inputs. *Electr Power Energy Syst* **58**, 246–256 (2014)
7. S. Ashraf, A. Gupta, D. Choudhary, S. Chakrabarti, Voltage stability monitoring of power systems using reduced network and artificial neural network. *Electrical Power and Energy Systems* **87**, 43–51 (2017)
8. P. Duraipandy, D. Devaraj, Extreme learning machine approach for on-line voltage stability assessment. *Swarm Evol Emotic Comput* **8298**, 397–405 (2013)
9. K.S. Sajan, V. Kumar, B. Tyagi, Genetic algorithm-based support vector machine for on-line voltage stability monitoring. *Electrical Power and Energy Systems* **73**, 200–208 (2015)
10. M. Suganyadevi, C. Babulal, Fast assessment of voltage stability margin of a power system. *J. Electr Syst* **10**(3), 305–316 (2014)
11. M. Suganyadevi, C. Babulal, Support vector regression model for the prediction of load-ability margin of a power system. *Appl. Soft Comput.* **24**, 304–315 (2014)
12. P. Duraipandy, D. Devaraj, On-Line Voltage Stability Assessment using Least Squares Support Vector Machine with Reduced Input Features, International Conference on Control, Instrumentation, Communication and Computational Technologies, Kanyakumari, India (2014)
13. M. Amroune, I. Musirin, T. Bouktir, M.M. Othman, The amalgamation of SVR and ANFIS models with synchronized phasor measurements for on-line voltage stability assessment. *Energies* **10**(11), 1693 (2017)
14. M. Amroune, I. Musirin, T. Bouktir, Power system voltage stability assessment using a hybrid approach combining dragonfly Optimization algorithm and support vector regression. *Arab J Sci Eng* **43**, 3023 (2018)
15. M. Amroune, Machine Learning Techniques Applied to On-Line Voltage Stability Assessment: A Review. *Arch Computat Methods Eng* (2019). <https://doi.org/10.1007/s11831-019-09368-2>
16. P. Modi, S. Singh, J. Sharma, Voltage stability evaluation of power system with FACTS devices using fuzzy neural network. *Eng. Appl. Artif. Intell.* **20**, 481–491 (2007)
17. P. Modi, S. Singh, J. Sharma, Fuzzy neural network-based voltage stability evaluation of power systems with SVC. *Appl. Soft Comput.* **8**, 657–665 (2008)
18. A. Berizzi, C. Bovo, D. Cirio, M. Delfanti, M. Merlo, M. Pozzi, Online fuzzy voltage collapse risk quantification. *Electr. Power Syst. Res.* **79**, 740–749 (2009)

19. P. Torres, H. Peralta, Castro a power system loading margin estimation using a neuro-fuzzy approach. *IEEE Trans. Power Syst.* **22**(4), 1955–1964 (2007)
20. S. Mohamed, M. El-Gaafary, A.M. Hemeida, Optimal power flow using moth swarm algorithm. *Electr. Power Syst. Res.* **142**, 190–206 (2017)
21. A. Parcheco-Vega, M. Sen, K.T. Yang, R.L. McClain, Neural Network Analysis of in-Tube Refrigerating Heat Exchanger with Limited Experimental Data. *Int. J. Heat Mass Transf.* **44**(4), 763–770 (2001)
22. D. Leila, K. Babak, K. Farshad, Optimal design approach for heating irregular-shaped objects in three-dimensional radiant furnaces using a hybrid genetic algorithm–artificial neural network method. *Engineering Optimization* (2017). <https://doi.org/10.1080/0305215X.2017.1323889>
23. P.P. Sarangi, A. Sahu, M. Panda, Training a Feed-Forward Neural Network Using Artificial Bee Colony with Back-Propagation Algorithm. (2014) In: Mohapatra D., Patnaik S. (eds) *Intelligent Computing, Networking, and Informatics. Advances in Intelligent Systems and Computing*, vol 243. Springer, New Delhi
24. Jang, Adaptive-network-based fuzzy inference system, *IEEE Trans-Syst Man Cybern.* **23**(3), 665–685 (1993)
25. S. L. Chiu, Fuzzy model identification based on cluster estimation. *Journal of Intelligent and Fuzzy Systems*, 267–278 (1994)
26. M. Seyedali, Moth-flame optimization algorithm: A novel nature-inspired heuristic Paradigm. *Knowl.-Based Syst.* **89**, 228–249 (2015)
27. R.N. Mantegna, Fast accurate algorithm for numerical simulation of Levy stable stochastic processes. *Phys. Rev.* **49**(5), 4677–4683 (1994)
28. S. Mirjalili, How effective is the Grey Wolf optimizer in training multi-layer perceptrons. *Appl Intell* **43**(1), 150–161 (2015)
29. M. Shahram, Optimal design of adaptive Neuro-fuzzy inference system using genetic algorithm for electricity demand forecasting in Iranian industry. Springer-Verlag, Berlin Heidelberg **20**, 4897–4906 (2016)
30. V. Salehi, M. Ossama, Real-Time Voltage Stability Monitoring and Evaluation Using Synchrophasor. *IEEE North American Power Symposium* (2011)
31. G. Yanfeng, N. Schulz, A. Guzman, Synchrophasor-Based Real-Time Voltage Stability Index. *Power Systems Conf, PSCE. IEEE PES*, 1029–1036 (2006)
32. M. Amroune, T. Bouktir, I. Musirin, Power system voltage instability risk mitigation via emergency demand response-based whale optimization algorithm. *Protection and Control of Modern Power Systems*, 4–25 (2019)
33. R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, Matpower: SteadyState Operations. *Planning and Analysis Tools for Power Systems Research and Education, Power Systems, IEEE Transactions on*, **26**(1), 12–19 (2011)
34. F. Milano, Power system analysis toolbox (PSAT), (Available at:) <http://faraday1.ucd.ie/psat.html>
35. M.N.M. Salleh, N. Talpur, K.H. Talpur, A Modified Neuro-Fuzzy System Using Metaheuristic Approaches for Data Classification, *Artificial Intelligence - Emerging Trends and Applications*, Marco Antonio Aceves-Fernandez. *IntechOpen* (2018). <https://doi.org/10.5772/intechopen.75575>
36. K. Gopalakrishnan, H. Ceylan, O. Attok-Okine, *Intelligent and Soft Computing in Infrastructure Systems Engineering* (Springer, Berlin, 2009)