



Article

Voltage Stability Margin Index Estimation Using a Hybrid Kernel Extreme Learning Machine Approach

Walter M. Villa-Acevedo ^{1,*} , Jesús M. López-Lezama ¹  and Delia G. Colomé ²

¹ Departamento de Ingeniería Eléctrica, Facultad de Ingeniería, Universidad de Antioquia, Calle 70 No 52-21, Medellín 050010, Colombia; jmaria.lopez@udea.edu.co

² Instituto de Energía Eléctrica, Facultad de Ingeniería, Universidad Nacional de San Juan, Avenida Libertador General San Martín 1109 (Oeste), San Juan 5400, Argentina; gcolome@iee-unsjconicet.org

* Correspondence: walter.villa@udea.edu.co; Tel.: +57-3007347553

Received: 23 January 2020; Accepted: 12 February 2020; Published: 15 February 2020



Abstract: This paper presents a novel approach for Voltage Stability Margin (VSM) estimation that combines a Kernel Extreme Learning Machine (KELM) with a Mean-Variance Mapping Optimization (MVMO) algorithm. Since the performance of a KELM depends on a proper parameter selection, the MVMO is used to optimize such task. In the proposed MVMO-KELM model the inputs and output are the magnitudes of voltage phasors and the VSM index, respectively. A Monte Carlo simulation was implemented to build a data base for the training and validation of the model. The data base considers different operative scenarios for three type of customers (residential commercial and industrial) as well as N-1 contingencies. The proposed MVMO-KELM model was validated with the IEEE 39 bus power system comparing its performance with a support vector machine (SVM) and an Artificial Neural Network (ANN) approach. Results evidenced a better performance of the proposed MVMO-KELM model when compared to such techniques. Furthermore, the higher robustness of the MVMO-KELM was also evidenced when considering noise in the input data.

Keywords: kernel extreme learning machine algorithm; machine learning techniques; near real time; voltage stability assessment; voltage stability index

1. Introduction

Currently, the competitive tendency of deregulated electricity markets, along with limitations in network expansion planning due to several factors that include environmental constraints and investment delays have caused electric power systems to often perform very close to their operative and stability limits. One of these relates to voltage stability, which is violated when the power system no longer has the capacity to maintain stable voltages in all or some of its buses after a disturbance has taken place. Furthermore, voltage instability has been recognized as one of the main problems in power systems around the world [1–3].

The ever-growing integration of intermittent renewable energies and the development of smart grids have increased the complexity of power systems operation and planning. In this context, conventional methods of voltage stability based on offline studies are not up to the challenges that are facing current power systems [4]; therefore, system operators must adopt new methods to monitor and evaluate voltage stability near real time. This necessity has motivated the development of robust and sophisticated tools for the supervision and evaluation of power systems. Monitoring of these systems is essential to guarantee a safe operation, and it must be carried out in real time to reach a high accuracy [5–7].

For near real time stability assessment, estimation time must be short and calculation efforts minimum. Stability performance indicators have traditionally been used to establish the distance

between the current operating point and a point at the voltage instability boundary. The main characteristic of these indices is that they can be estimated when the operating conditions change. In addition, these indices must be predictable and quickly calculable [8,9]. The problem with most stability indices is that their values vary in a highly nonlinear and discrete way, due to the non-linear characteristics of the system and its operating limits [8–11]. Other indices require a highly intensive calculation to determine their value for changing system conditions. Another important aspect to reach near real time assessment is to improve measurement of key power system variables; this has been recently achieved with Phasor Measurement Units (PMUs) that allow observing fast dynamic behaviors due to their high sampling rate.

In the last decades, the use of Artificial Intelligence (AI) tools has arisen as a solution for the evaluation of voltage stability near real time. The stability evaluation using AI drastically reduces the calculation time, eliminating the need of computing the nonlinear equations of the system models. AI techniques capture the relationships among the different states of the system and system stability information, extracting knowledge from these data and determining the corresponding stability status [4].

Voltage Stability evaluation using AI tools requires building a learning database in offline mode with data of many operational conditions. This database must contain the electrical variables that allow determining the actual condition of the power system in near real time. When the AI is trained, new information is fed to the AI; being the output a voltage stability indicator [12,13]. The following AI techniques using voltage stability assessment are reported in the scientific literature: Artificial Neural Network (ANN), Support Vector Machine (SVM) and Extreme Learning Machine (ELM).

ANNs have been widely used to estimate voltage stability indicators in both long and short term voltage stability assessment, they derive their computer power through a parallel-distributed structure that allows them to learn and generalize. One of the main features that an ANN provides for power system applications is their ability to deal with complex non-linear mapping through a set of input/output data. In the case of voltage stability studies the ANN identifies the existing relationships between an input set as measurable parameters and the output that is the stability indicator [14–16]. System parameters such as nodal voltage magnitudes and angles are used as the ANN input variables. These input variables present lower errors in the voltage stability indicator estimation [17]. Several ANN approaches based on backpropagation [18,19], Kohonen [20], Radial Basis Function (RBF) [21] and ANN with reduced input features have been reported in the specialized literature [22,23]. A large number of operating conditions has to be generated for the training and testing when ANNs are used. Thus, the power system is simulated with distinct operating conditions, including sometimes contingencies. These simulated conditions are used to obtain other system parameters such as branch currents, line flows, real and reactive power and different voltage stability indicators such as line voltage stability indices, bus voltage stability indices and Voltage Stability Margin (VSM) or Loading Margin (LM) [24–26]. Although ANNs have been extensively used by researchers, they exhibit some shortcomings, mainly related to the long time required in their training process, and the fact that its learning is highly dependent on the number of training data.

SVM is a machine learning technique based on statistical learning theory and structural risk minimization. In recent years, support vector regression (SVR), the regression version of SVM has been applied to voltage stability monitoring [27,28]. The proposed SVR method allows estimating the loading margin of the system in normal operation conditions and considering different demand growing scenarios. In [27] the contingencies that affect the LM estimation of the system are not considered; also the input vectors for the SVR, such as the phasor voltage of the system nodes that are delivered by the PMUs are not considered. On the other hand, in [28] the Power Transfer Stability Index (PTSI) is considered as the output stability index that requires impedance information and Thevenin voltage. In this case, dynamic voltage collapse prediction is determined based on the PTSI. The data collected from a dynamic simulation is used as input to the SVR which is used as a predictor to determine the dynamic voltage collapse indices of the power system.

In [29] a SVR is proposed for online voltage stability evaluation with a reduced set of input variables. The vector dimension is reduced using Principal Component Analysis (PCA) combined with a feature selection technique called Mutual Information (MI). The VSM is used to complete the voltage stability evaluation. In this case, only the most severe contingencies are considered in the simulation scenarios, impacting negatively the VSM of the system. Voltage phasor system nodes are not considered and the selected input variables are the active and reactive powers of load nodes.

An inconvenience that arises when working with SVR is the accurate selection of its internal parameters in relation to any information of the problem. This allows ensuring a high accuracy and good performance of the AI. An inadequate selection of the SVR parameters leads to overfitting or under fitting. In [30], a metaheuristic known as Artificial Immune Least Square is used for obtaining the optimum parameters of the SVR, that executes the voltage stability index estimation. The index named as Voltage Stability Condition Indicator (VSCI) is used as a tool to evaluate the stability at load nodes, and the SVR is used to estimate the VSCI. For its estimation, both active and reactive powers are needed; however, the measures of the voltage phasor are not considered. Other metaheuristic optimization techniques used for the selection of the SVR parameters include Genetic Algorithms (GA) [31], Particle Swarm Optimization (PSO) [32] and Dragonfly Optimization algorithm (DFO) [33].

In [31] a GA is used to improve the accuracy and performance of the SVR, that completes the stability index estimation. Complex voltages of buses that are given by the PMUs are used as inputs and the output corresponds to the VSM. In the generated scenarios, the occurrence of contingences and their impact in the voltage stability are not taken into account. In [32], a PSO determines the setting of the SVM parameters. The proposed alternative for the stability evaluation uses as input vector for the SVR the voltage angles and the reactive power load; while the VSM index is used as an output vector. Similarly, in [33] a voltage stability evaluation technique based on swarm intelligences is proposed, as well as a DFO for the setting of the SVR parameters, aiming at completing the evaluation of the online voltage stability. In this case, the PMUs voltage magnitudes are used as an input vector for the hybrid DFO-SVR, and the output vector corresponds to the minimum values of the voltage stability index.

Extreme Learning Machine (ELM) is a new machine learning tool that is being applied for online stability evaluation. ELM has shown better performance and lower training times than ANN and SVM facing regression problems [34]. In [35–37], ELM is used as a tool to evaluate the long-term online voltage stability, considering different electric parameters as input vectors. These include voltage magnitudes and angles, power flows, as well as active and reactive power injections. In [38], an assembled ELM is proposed to estimate the voltage stability of a power system. The proposed method improves the performance of the VSM estimation, but increases the training times for the ELM set.

In this paper, a long-term online voltage stability evaluation is proposed through a VSM index using a novel AI technique called Kernel Extreme Learning Machine (KELM). KELM is a combination of ELM with an AI Kernel type, improving the performance and keeping a short training time. To guarantee a good performance of the KELM, the Mean-Variance Mapping Optimization algorithm (MVMO) is proposed for the selection of the internal KELM parameters. A comparison of the performance of the hybrid MVMO-KELM model with SVM and an ANN is presented, and an analysis of cases with the 39-node IEEE testing system is performed. Results evidenced the robustness and effectiveness of the proposed MVMO-KELM model.

2. Voltage Stability Assessment

Voltage stability is concerned with the ability of the power system to maintain acceptable voltages both under normal conditions and under disturbances [7]. Voltage stability assessment deals with finding the voltage levels of the system under different loading conditions to ascertain the stability limit. Voltage stability is analyzed by employing two practices, namely time-domain (dynamic) simulation

and steady state analysis. Depending on the stability phenomenon or phenomena under investigation, one or both of these techniques may be applied.

In the long-term voltage stability, the system dynamics are usually slow. So, many aspects of the problem can be effectively analyzed by using static methods, which allow examinations of a wide range of system conditions and provide insight into the nature of the problem and identifies the key contributing factors. For static voltage stability analysis, the loading of the system is slowly increased (in certain direction) to the point of voltage instability frontier.

Several indexes have been proposed to evaluate long-term voltage stability in power systems [8–11]. The general idea of a voltage stability index is to define a scalar magnitude that determines the distance from the current operation point to the voltage instability limit. It is important that voltage stability indexes are sensitive to loading changes and present a predictable behavior in relation to its own increases. The latter allows extrapolating the additional power that might be supplied by the power system before getting into the instability voltage point. It is also relevant that the stability indexes can be monitored when the system parameters change, and that their calculation is fast enough so that online system supervision is feasible.

Among the most used and accepted indexes for long-term voltage stability assessment, there are the deviation indexes category. These indexes require an increment of the system demand; from its current state, the load is increased following a previously defined pattern to reach the instability point. In general, these indexes provide a distance to the instability point in MW/MVAR. An example is the LM which is the most basic and widely accepted to estimate the distance to the instability frontier. The LM is defined as the distance in terms of power (loading parameter, λ) between a current operation point (base case, λ_0) to the instability voltage point (see Figure 1). This indicator takes into account the nonlinear behavior that occurs between the base case and the instability boundary.

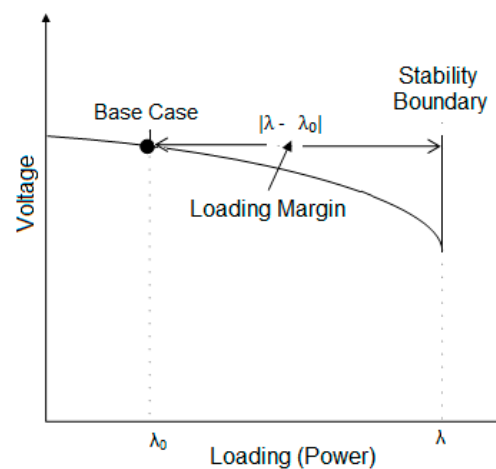


Figure 1. Definition of loading margin.

The advantages of LM as a voltage stability index are: (a) The static model of the power system is required, it can be used with dynamic systems, but it does not depend on dynamic details; (b) it is an accurate index that takes into account the nonlinear characteristics of the system and its limits, such as the reactive power limits that increase as the demand increases; (c) limits do not directly reflect sudden changes on the LM; and (d) the LM considers the demand increase patterns, that is, increasing load directions in the parametric demand space [8].

Among the disadvantages of the LM as a voltage stability index, the following should be considered: (a) Evaluations in operating conditions away from the current operation point are required, that is why the computational demands are higher than in the calculations of indexes that are based on using only the information of the current operation point. This computational cost would be the

most serious disadvantage of the LM. (b) A demand increase direction is required; nevertheless, this information is usually easily acquired.

Some specific methods to increase the speed and reliability of the calculations of the voltage stability limits have been elaborated. These are the Continuation Power Flow (CPF) method [39], and the collapse point method [8].

3. Artificial Intelligence (AI) Systems

The advantages that make AI a promising alternative for the estimation of a voltage stability margin index include [4]:

- (a) Fastness: using a trained AI, it is possible to obtain a value of the stability system margin in the current operation point. The AI determines the margin in a fraction of a second after receiving the input data, which allows an appropriate response to prevent instability.
- (b) Knowledge extraction: the AI can extract stability system information; this provides an understanding of the system operation.
- (c) Less data capacity: for the conventional evaluation methods, accurate information is required as well as a complete description of the system. In contrast, the AI evaluates the stability only with the available and significant parameters.
- (d) Generalization capacity: the AI simultaneously handles a wide spectrum of scenarios or system conditions in the stability evaluation; these conditions can be previously assumed and not foreseen.

In the following section, a brief description of the AI techniques that have been used in the evaluation of voltage stability is presented.

3.1. Artificial Neural Network (ANN)

An ANN is an AI technique composed of a great number of processing elements highly interconnected (artificial neurons) working at the same time to solve a specific problem. The artificial neuron is a processing information unity that finds nonlinear relations among sets of data. They are called ANN because they learn by experience; this is given by system training with experimental data so the network acquires the knowledge related to the problem being studied. An ANN is composed of many simple elements that work in parallel. The network design is mostly determined by the connections among its elements [40]. A neural system imitates the brain in two aspects; (a) knowledge is acquired by the system through a learning process, (b) connections among the neurons, and known as synaptic weights, are used to store knowledge.

3.2. SVM

SVM is a type of machine learning technique used in solving problems of classification, regression, and pattern recognition [41]. SVM is a promising method for solving problems of linear and nonlinear classification. A SVM is an algorithm that uses a nonlinear mapping to transform original training data to a larger dimension. In this new dimension the SVM searches the hyperplane of optimal linear separation (decision limit) splitting one type from another. With an appropriate nonlinear mapping to a highly enough dimension, data of both types are always possible to be separated by a hyperplane. The SVM finds this hyperplane using support vectors and margins (defined by the support vectors); this is achieved through solving an optimization problem [42]. SVM belongs to a set of algorithms named Kernel-based methods and the structural risk minimization is used as the optimization principle [43]; also, SVM has statistical robustness and ability to overcome over adjustment problems [44].

SVR is an extension of the SVM with a good generalization capacity created to solve time series prediction problems and function approximation (regression). The function approximation consists on determining the relation between the inputs and outputs using data pairs corresponding to inputs and outputs (x_i, z_i) where $i = 1, \dots, l$; x_i is the input vector row i of n dimension, z_i is the output scalar

row i , and l is the number of training data [45,46]. The SVR assigns the input space to a space of multidimensional characteristics to determine an optimal hyperplane defined as:

$$f(x) = w^T \phi(x) + b \quad (1)$$

where w is a pondered n -dimensional vector, $\phi(x)$ corresponds to the mapping functions from space x to the feature space, and b is the threshold term [46].

This work uses the support vector machine for regression problems called ε -SVR with a radial base function (RBF) as kernel. For a set of training points (x_i, z_i) where $x_i \in R^n$ is a feature vector and $z_i \in R^1$ is an objective output vector, with given parameters $C > 0$ and $\varepsilon > 0$; the standard form of ε -SVR is given by (2)–(5) [47]:

$$\min : \frac{1}{2} w^T w + C \sum_{i=1}^l \xi_i + C \sum_{i=1}^l \xi_i^* \quad (2)$$

$$sa : w^T \phi(x_i) + b - z_i \leq \varepsilon + \xi_i, \quad (3)$$

$$z_i - \phi(x_i) - b \leq \varepsilon + \xi_i^*, \quad (4)$$

$$\xi_i, \xi_i^* \geq 0, i = 1, \dots, l. \quad (5)$$

where C is the margin parameter that determines compensation between the margin magnitude and the estimation error of the training data, ξ_i and ξ_i^* are slack variables associated to x_i [45]. The mapping function $\phi(x)$ is called kernel function, and the RBF is used for the SVR like kernel that is defined as:

$$K(x_i, x_j) = \phi(x_i)^T \phi(x_j) = e^{-\gamma \|x_i - x_j\|^2}, \gamma > 0 \quad (6)$$

where γ is the kernel parameter, this must be determinate. On the other hand, a small positive parameter ε is used to reduce residual errors in the regression, that is why an interval linear function is proposed rather than a squared error function [45]. In (3) and (4) the parameter $\varepsilon > 0$ corresponds to the radius of a zone that is named the ε insensibility zone or tube; the ideal estimation is achieved when the training data are in this zone. Equations (3) to (5) allow the existence of data outside the tube, hanging non-negative variables ξ_i and ξ_i^* are used. A more detailed explanation regarding the ε -SVR can be consulted in [45]. The C and ε parameters in (2) and γ in (6) must be defined before the SVM, other parameters are determined in the optimization process defined in (2) to (5).

3.3. KELM

One of the inconveniences of ANNs relates to their low training speed, which has been the main obstacle for applications in which speed is of paramount importance. This is because most ANN training algorithms are based on descending gradient methods and all network parameters are synchronized iteratively in the training.

A new learning algorithm called Extreme Learning Machine (ELM) was proposed by Huang et al. in 2006, which is used both in classification and regression problems. The main advantage of ELMs is the absence of iterative adjustment in the hidden layer of the network; therefore, its training time is lower than those of traditional ANNs [48].

For ELMs, it is proved that the input weights and the hidden layer threshold of a Single Layer Feedforward Network (SLFN) can be randomly assigned. If these activation functions in the hidden layer are infinitively differentiable, the SLFN can simply be considered as a linear system and the output weights (hidden layer connections to the output layer) of the SLFN can be analytically determined through a simple opposite generalized operation of the output matrices of the hidden layer output [34].

The ELM training algorithm follows the following steps: (1) randomly assign hidden node parameters: input weights w_i and threshold b_i ; (2) calculate the hidden layer output matrix H , and (3) obtain an output weight vector β . Given a set of N input vectors samples (x_j, t_j) , and standard

SLFNs with n hidden nodes whose weights are randomly initialized, and activation functions $g(x)$ are modeled as:

$$\sum_i^n \beta_i g(w_i x_j + b_i) = t_j, \quad j = 1, \dots, N \quad (7)$$

where w_i is the weight vector connecting the i th hidden node and input nodes; β_i is the weight vector connecting the i th hidden nodes and the output nodes, b_i is threshold of i th hidden node while t_j and x_j are j th target vector and input vectors, respectively. Equation (7) can be expressed in a matrix form as indicated in (8):

$$H\beta = T \quad (8)$$

where H is the output matrix ($N \times n$) of the hidden layer with respect to the input x_j , β is the output weight matrix ($n \times m$), T is the target matrix ($n \times m$) and m is the number of distinct arbitrary targets. ELM training consists of a minimum norm least-squares solution of a general linear system, obtaining optimal weights $\beta = H^+T$, where $H^+ = (H^TH)^{-1}H^T$ is the Moore–Penrose generalized inverse [48]. The final expression for Equation (8) is:

$$\beta = H^T \left(\frac{I}{\lambda} + HH^T \right)^{-1} T \quad (9)$$

According to the ridge regression theory a positive value $1/\lambda$ is added to the diagonal of HH^T [34,49]. The corresponding output function of ELM is given by (10), where $h(x)$ is the hidden layer feature mapping function.

$$f(x) = h(x)\beta = h(x)H^T \left(\frac{I}{\lambda} + HH^T \right)^{-1} T \quad (10)$$

In the KELM algorithm, the hidden layer feature mapping function $h(x)$ is known, but can define a kernel function $K(u, v)$ to find the value of the output function. The kernel function directly adopts the form of an inner product. Additionally, it is not necessary to set the number of hidden layer nodes when solving the output function, so that the initial weight and threshold of the hidden layer do not need to be set [50,51]. A kernel function is introduced in KELM to obtain better regression accuracy, as:

$$f(x) = h(x)H^T \left(\frac{I}{\lambda} + HH^T \right)^{-1} T = \begin{bmatrix} K(x, x_1) \\ \vdots \\ K(x, x_N) \end{bmatrix}^T \left(\frac{I}{\lambda} + \Omega_{ELM} \right)^{-1} T \quad (11)$$

$$\Omega_{ELM}(i, j) = h(x_i) \cdot h(x_j) = K(x_i, x_j) \quad (12)$$

where Ω_{ELM} is the kernel function matrix, $K(u, v)$ is the kernel function, as a kernel, the Gaussian function is usually chosen (see Equation (6)); finally, N is the input layer dimension.

4. Proposed Methodology

A long-term online voltage stability evaluation is proposed in this paper through the VSM index using KELM. The steps of the methodology are detailed in this section.

4.1. Building of AI Learning Database Using Monte Carlo Method

A Monte Carlo (MC) simulation is proposed to build different pre and post contingency operative conditions close to real life operation. The operative conditions consist of different scenarios or hourly demand curves (with a 24-h horizon) that are transferred to the nodal demands of the PQ buses. The scenarios are built from the demand of the base case, taken from the original information of the system. Then, the demand is changed in every node using hourly demand according to every

type of user as indicated in Figure 2 (residential, industrial or commercial) [43]. Every PQ node is randomly assigned a type of user, guaranteeing the same number of PQ users for each type of user. The uncertainty in the demand is considered using a probability density function (PDF) of a normal distribution.

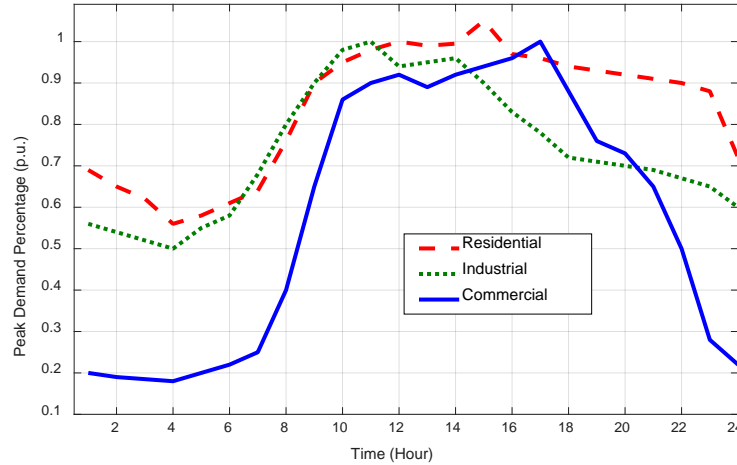


Figure 2. Daily hourly demand curves according to the type of user [43].

Every outage (N-1 contingency) is considered as an independent and randomly generated event. The selection of contingencies depends on the probability of forced outage of the system components. The probability of a contingency can be estimated using the Poisson distribution with a constant occurrence rate [52]. Using the Poisson distribution formula, the probability of occurrence of a N-1 contingency in a given period of time is given by the following equation:

$$P_0 = 1 - e^{-\lambda_0 t} \quad (13)$$

where: λ_0 is the average occurrence rate of a contingency related to an element of the system, and t is the duration considered; in this case, it lasts one hour.

Equation (13) is applied to all components that are considered in the eligible set of N-1 contingencies. In this case, contingencies caused by the output of transmission lines, transformers and generation units are considered. For probabilistic contingency selection using the MC Method, it is assumed that each component of the system has two operating states: normal and failure status. In the MC simulation, a uniformly distributed random number, R , is generated for each component or group of components, the Markovian model is handled as follows [53]:

$$I_i = \begin{cases} 1 & R_i > P_0 & \text{Normal status} \\ 0 & R_i \leq P_0 & \text{Failure status} \end{cases} \quad (14)$$

where I_i is the operating state of the i th component of the system, R_i is a random number distributed for the i th component. For the elements in a failure state, the contingency analysis is performed, which corresponds to executing a power flow to determine the post-contingency conditions of the system. The proposed MC process flowchart to build the database for the AI training stage is presented in Figure 3.

The MC method starts with the generation of a random sample from the PDFs of the input variables considered in the process, for example, the demand in each node, and the random selection of the contingency in an element. Then, an Optimal Power Flow (OPF) is executed for each sample of the input variables in order to define a stable pre-contingency state scenario. A static N-1 contingency analysis and a Continuation Power Flow (CPF) are run to determine the VSM using the UWPFLOW

program [54]. Finally, pre and post contingency power flow outcomes and the VSM corresponding to each evaluation carried out in the MC process are obtained.

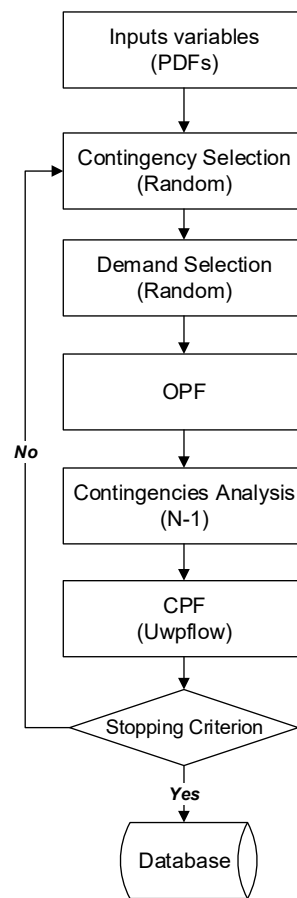


Figure 3. Flowchart of the Monte Carlo (MC) process to build database.

With the CPF tool, the VSM is calculated as global index. In this case, that corresponds to the maximum point of transfer in the PV curve [55]. The distance from the system operating point to the voltage instability frontier depends on the load increase trajectory and the initial operating point of the system. In addition, the occurrence of N-1 contingencies affects the stability frontier, that is, the distance from the operation point to the instability frontier or VSM is reduced. In the MC process, different load increase directions are considered. This is achieved by varying the operating points and retaining a constant power factor for that operating point when the demand is increased.

Figure 4 shows different operating points as a function of the real and reactive power demand, and how load increase directions and contingencies affect the VSM.

4.2. AI Training

The database for AI training and validation is formed by two matrixes $X_{N \times n}$ which are composed of the features and $Y_{N \times 1}$ target values. The feature matrix is formed by N observations that correspond to the voltage phasor magnitudes of the n buses of the test power system; and target values that correspond to the VSM of the N different system conditions that are represented by voltage phasor magnitudes, that is, N snapshots of the power system dynamics.

The AI training stage is done in an offline mode, in which the common strategy is to separate a set of training data and other set of testing data (unknown data). These unknown data are used to evaluate the estimation accuracy (validation process). The objective of the training stage is to build an AI model (SVM and KELM) based on the immerse information of the training set [42].

Before executing the training of the type kernel learning machine techniques, the optimal parameters of the AI must be found in order to improve the adaptation of the training data and the accuracy of the AI response facing unknown data [46].

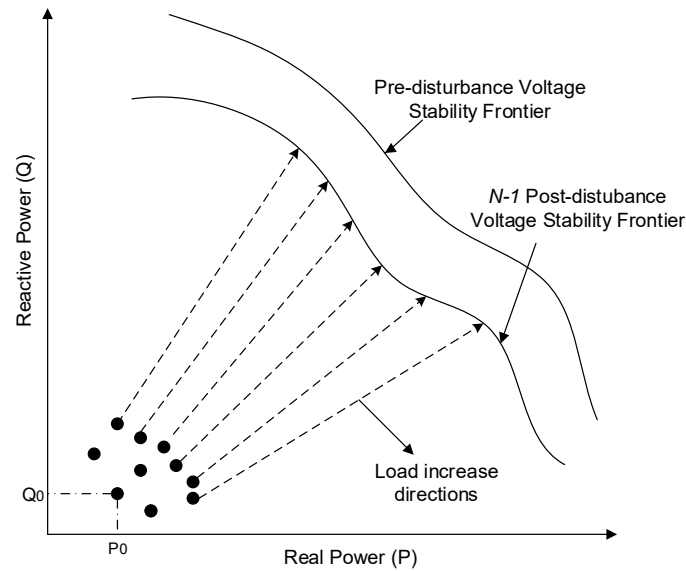


Figure 4. Voltage Stability Margin (VSM) variation due to different operation points, load increase directions and N-1 contingencies.

Generally, SVR with kernel type RBF has three parameters to optimize: C , γ and ε . For these SVR parameters selection, two methods have been combined: (i) a k -folks cross-validation based on k repetitions [42,46], that allows preventing an overfitting problem, and (ii) the grid search method, that allows obtaining the AI parameters.

In the k -folk cross-validation, the data set is randomly divided into k subsets, which are mutually exclusive and have the same size. Both training and testing are performed k times, where a testing subset is kept and the remaining $k - 1$ subsets are the training data. According to [42,46] it is recommended is to use the grid search algorithm using cross-validation, where several sets of (C, γ, ε) are tested and those with the lowest error prediction in the cross-validation are chosen. The grid search algorithm is a procedure of high computational burden that takes more calculation time according to the number of pairs considered; furthermore, this procedure does not guarantee getting the optimal AI parameters.

4.3. Determination of AI Optimal Parameters

As a solution to the optimal parameter setting, different metaheuristic optimization techniques have been proposed, such as Particle Swarm Optimization [56], Genetic Algorithms [31] and Mean-Variance Mapping Optimization (MVMO), where an objective function (OF) is minimized and corresponded to the Mean Squared Error (MSE) [43]. MVMO has successfully been applied for the solution of different power system optimization problems. Furthermore, Numerical comparisons between MVMO and evolutionary algorithms have shown that MVMO exhibits a better performance, especially in terms of convergence speed [57–59]. In this case, the MVMO tool is used and the OF is defined as:

$$OF = \sum_{i=1}^k \left(\frac{|t - y|^2}{n} \right)_i \quad (15)$$

Subject to:

$$x_{min} \leq x \leq x_{max}$$

where t is the desired output vector, y is the AI prediction vector, $|\cdot|$ corresponds to the Euclidian distance and n corresponds to the y vector dimension. The sub-index i refers to the i th-iteration of the k -folks cross-validation process, and the vector x comprises the AI parameters that solve the problem.

The flowchart of the proposed methodology for the identification of the AI parameters using MVMO is presented in Figure 5. First, the AI training intelligence is organized; then, each AI parameter is initialized; the MVMO proposes the parameters to optimize, executes the optimization process and delivers the optimum parameters found, with which the MSE is calculated. For each k -folk cross-validation run, the MVMO minimizes the prediction error (PE_i), being i the i th run for the validation run. At the end of the process, the MVMO delivers the optimum parameters of the AI that minimize the total error of all the runs executed in the cross validation.

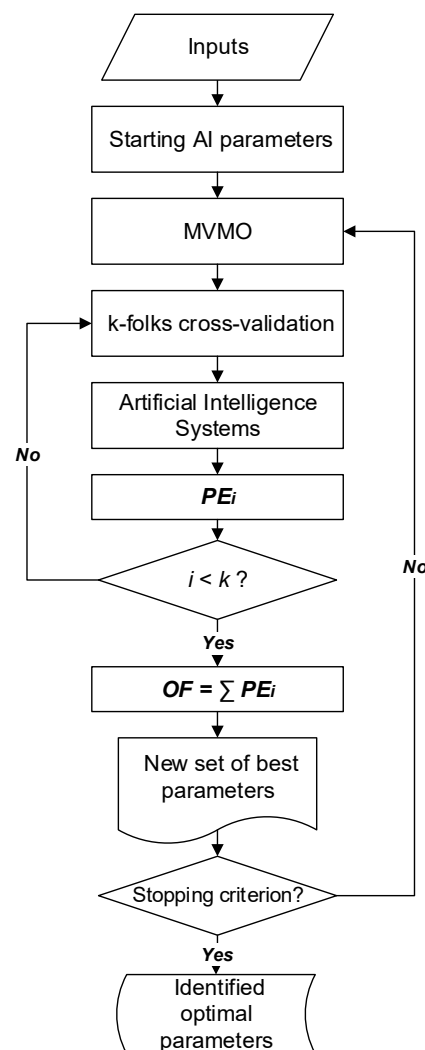


Figure 5. Methodology for identification of Artificial Intelligence AI parameters based on Mean-Variance Mapping Optimization (MVMO).

Before training both SVR and KELM, the optimum parameters that allow getting a better accuracy in the AI response must be determined [34,46]. As mentioned before, k -folk Cross-Validation is used with the “grid search” method to prevent overfitting problems. This procedure is computationally intensive given that the number of the parameters combination is huge. Additionally, it is a process of local search whose search range is hard to establish. As a solution to the above mentioned problems, an alternative presented in [43] was developed, there an optimization heuristic algorithm known as MVMO is used to determine the best parameters of the AI with kernel. The optimum parameters that

must be identified for SVR are ε , C and γ according to equation (2); while the optimum parameters to identify for KELM are C and γ .

5. Test and Results

5.1. Database for AI Training

In order to evaluate the performance of the different AI techniques under study, the database for the IEEE 39-bus test system (also known as the New England IEEE test system) is built [60]. The test system (depicted in Figure 6) comprises 10 generators with active and reactive power limits; 29 load or PQ buses, 12 transformers, and 34 transmission lines. The data of the transmission lines, transformers, nodal demand, active and reactive power limits were taken from [61].

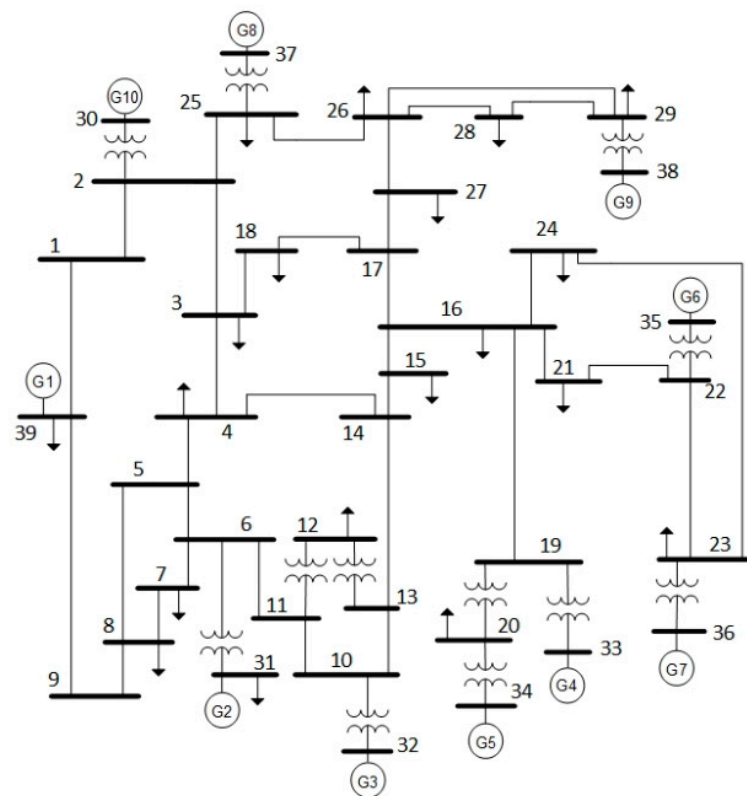


Figure 6. Single-line diagram of New England 39 bus test system [60].

The MC process was implemented with the following parameters. For each nodal demand of the PQ buses, the means are taken as the active and reactive power of the original information (provided in [61]) and deviations of 8% are considered for the nodal demand in each PQ bus. Furthermore, the N-1 contingencies are generated, assuming an average outage occurrence rate equal to 0.04, both for transmission lines and generation units.

The static CPF tool is run for every contingency of the system. This is done for the whole system or with contingencies in agreement to the respective case. The VSM is determined for both pre and post-contingency conditions in the test system. An input matrix $X_{N \times n}$ is formed by 20000 operative conditions (normal and under contingency), where N corresponds to the number of simulated operative conditions (snapshots) and n corresponds to the voltage phasors of buses taken for each simulated operative condition. In this case, the voltage phasors are 39 values that correspond to the number of system buses. Additionally, an output $Y_{N \times 1}$ matrix is built in order to store the calculated VSM for each simulated condition that is evaluated.

In this case, 70% of the data was used for training and 30% was used in the AI validation. An important aspect prior to the training of intelligent systems, especially the AI with kernel, was the need to identify the optimum kernel parameters that present better results for the data handled by each AI [46]. The parameters identification method of the AI is presented in the next section.

5.2. AI Optimal Parameters Selection

The proposed methodology for the identification of AI parameters, using MVMO, is presented in Section 4.3. In this case, 10 repetitions are used in the k-fold cross-validation to ensure a robust regression. Based on [43], the search space of the AI parameters are defined in an exponential scale in the following way: $C \in [2^{-5}, 2^{15}]$, $\gamma \in [2^{-15}, 2^5]$, $\varepsilon \in [2^{-5}, 2^0]$ [42,46].

A library for support vector machine (LIBSVM) was used to execute the SVR for the design, training and testing model of the SVR [46]. For a more detailed description, the computational code adapted in this work and used to execute the KELM, can be consulted in [62]. The neuronal network package from Matlab [63] was used to execute the ANN (backpropagation type). The number of neurons in the hidden layer was chosen empirically; this one was varied to guarantee a good performance and reduce the time in the training process.

The parameters identification is applied to the generated data of voltage phasor and VSM using the 39-bus test system. Figure 7 illustrates the convergence of the MVMO, which was implemented to identify the optimum parameters for both SVR and KELM training. Note that the convergence process of the MVMO to find the KELM optimum parameters is faster than the one for the SVR, requiring only a few number of objective function evaluations.

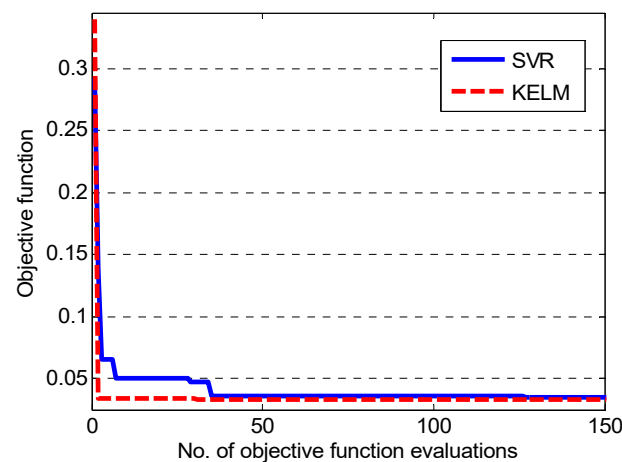


Figure 7. Convergence of the MVMO in the process of identifying optimal parameters of the AI.

Table 1 presents the optimal parameters for each AI considered. The approximation of functions (AI-regressors) were obtained for 15,000 samples performed using the MC process for the test system.

Table 1. Optimal identified parameters for each AI.

AI-Regressor	Identified Parameters		
	$\text{Log}_2 C$	$\text{Log}_2 \gamma$	$\text{Log}_2 \varepsilon$
SVR	12.888	0.675	−4.778
KELM	7.840	−5.237	—

5.3. Comparison Performance of Machine Learning Techniques in VSM Estimation

Several tests were considered to evaluate the performance of the VSM estimation for each AI technique under study. The general test cases consist in that for each simulated operating condition,

the information of the voltage phasors of all buses and their respective VSM are generated. The AI system is trained with the optimal parameters obtained in the previous section. For the performance test of the AI, the following conditions are considered [64]:

- Case 1: Voltage phasors measurements without noise.
- Case 2: Voltage phasors measurements with noise in the phasor magnitude. The noise is added randomly following a normal distribution with zero mean and deviation equal to 0.01 p.u.
- Case 3: Voltage phasors measurements with noise in the phasor magnitude, and with zero mean and deviation equal to 0.04 p.u.

Table 2 shows the results performance of the AI considering several conditions of the test system that includes measurements with noise in the phasor magnitude. The Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) are defined as the performance indices for the estimates made by the AI-regressor. In this case, 5000 randomly samples were considered on the database for calculating MSE and RMSE.

Table 2. Obtained Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) in the AI test.

AI-Regressor	MSE			RMSE		
	Case 1	Case 2	Case 3	Case 1	Case 2	Case 3
ANN	0.0008	19.3113	69.5482	0.0291	4.3945	8.3396
SVR	0.0015	0.2909	1.7169	0.0389	0.5394	1.3103
KELM	0.0005	0.0417	0.0460	0.0242	0.2043	0.2144

The MSE and RMSE are calculated considering all the test samples. The training time corresponds to the training of 15,000 samples (different from the ones of the testing stage) while the testing time corresponds to the execution of the remaining 5000 samples. For the ANN, an optimal number of twenty neurons was identified for the hidden layer, and the optimal parameters of SVR and KELM are shown in Table 1.

In Table 2, it is observed that KELM has a more accurate prediction in normal conditions (see Case 1) than in conditions with high level of noise; situation that occurs in Case 2 and Case 3. It can also be noted in Table 2 that the second most accurate approach is ANN, followed by SVR.

Table 3 presents the training and testing time for each AI execution process. In this regard, it can be observed that KELM presents significantly lower training time than the other AIs (see second column of Table 3). The average execution time of KELM for 5000 test samples was 1.258 s, which is still adequate for the long-term voltage stability evaluation in near real time. CPF average execution time for a single run was 0.3216 s. Note that the KELM execution time is much shorter than the one of the CPF.

Table 3. Training and testing time for AI execution process.

AI-Regressor	Training Time (s)		Testing Time (s)	
	Case 1	Case 1	Case 2	Case 3
ANN	430.8661	0.0401	0.0187	0.0173
SVR	116.8658	1.3095	1.2880	1.3120
KELM	13.0696	1.2661	1.2774	1.2330

The predicted values of VSM by each AI model in the testing phase are plotted in Figures 8–10, the VSM is scaled between 0 to 1, and a small test set (about 55 samples) was selected to present in the figures.

Figure 8 displays the most accurate results. Note that the MVMO-KELM model provides results that are much more in correspondence with actual values of VSM for Case 1. The second most accurate results can be seen in Figure 9, which are obtained by using ANN model. Lastly, Figure 10 presents the results obtained by using MVMO-SVR model.

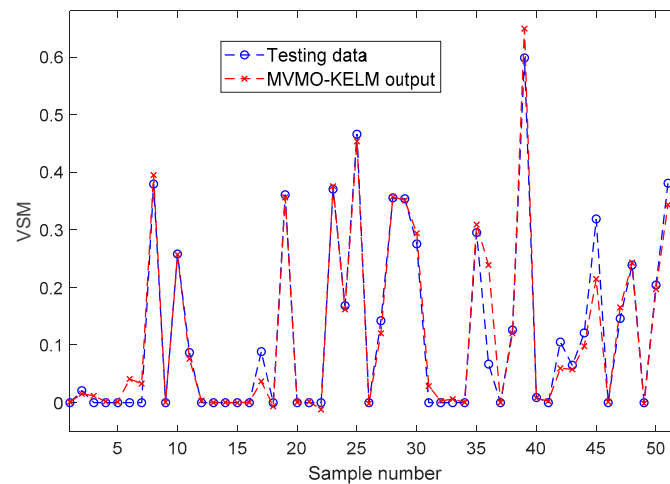


Figure 8. Comparison between actual and predicted values of VSM with MVMO-Kernel Extreme Learning Machine (KELM).

Case 1

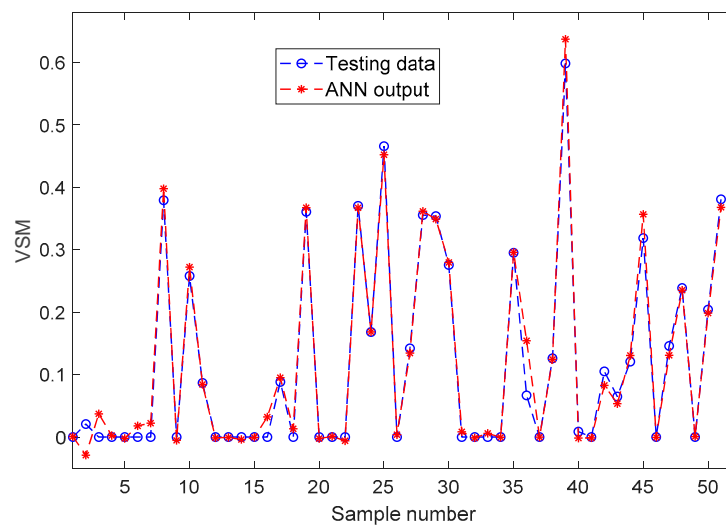


Figure 9. Comparison between actual and predicted values of VSM with Artificial Neural Network (ANN).

Case 1

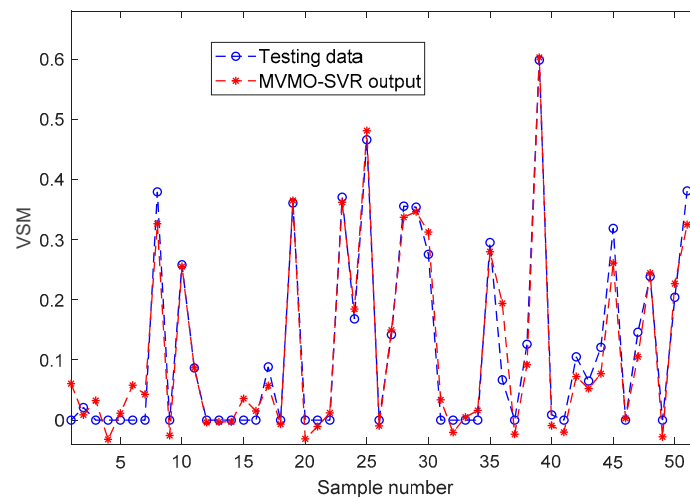


Figure 10. Comparison between actual and predicted values of VSM with MVMO-Support Vector Regression (SVR).

Case 1

6. Conclusions

This paper presented a hybrid AI model for the online estimation of VSM. The proposed hybrid model combines the strengths of KELM and MVMO. In this case, the MVMO is used to optimize the parameter settings of the KELM for the online estimation of voltage stability. The model was trained considering several operative conditions including different generation-demand scenarios with three types of consumers (commercial, residential and industrial) as well as N-1 contingencies.

The MVMO-KELM model was successfully implemented to estimate the VSM in a benchmark power system, demonstrating that the MVMO optimization technique guarantees an appropriate selection of the KELM parameters that improves the accuracy of the prediction. The performance of the MVMO-KELM model was compared, with the MVMO-SVR and an ANN. Different test cases were taken into account corresponding to the inclusion of several noise levels in the magnitudes of nodal voltages (inputs of the model). It was confirmed that the proposed MVMO-KELM model has a more robust performance against noise in the input data than that of the MVMO-SVR and ANN.

The proposed MVMO-KELM model can also be incorporated within a long-term voltage stability monitoring method. The MVMO-KELM model combined with other monitoring techniques, might constitute a powerful tool for the evaluation of voltage able to withstand several conditions in real-scale power systems.

Author Contributions: Conceptualization, W.M.V.-A., J.M.L.-L. and D.G.C.; data curation, W.M.V.-A.; formal analysis, W.M.V.-A., J.M.L.-L. and D.G.C.; funding acquisition, W.M.V.-A., J.M.L.-L. and D.G.C.; investigation, W.M.V.-A., J.M.L.-L. and D.G.C.; methodology, W.M.V.-A.; project administration, J.M.L.-L. and D.G.C.; resources, W.M.V.-A. and J.M.L.-L.; software, W.M.V.-A.; supervision, J.M.L.-L. and D.G.C.; validation, W.M.V.-A. and D.G.C.; visualization, W.M.V.-A. and J.M.L.-L.; writing—original draft, W.M.V.-A.; writing—review and editing, J.M.L.-L. and D.G.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Colombia Scientific Program within the framework of the so-called Ecosistema Científico (Contract No. FP44842- 218-2018).

Acknowledgments: The authors also acknowledge the financial support provided by the Colombia Scientific Program within the framework of the call Ecosistema Científico (Contract No. FP44842- 218-2018).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Liscouski, B.; Elliot, W. Final report on the august 14, 2003 blackout in the United States and Canada: Causes and recommendations. *Rep. US Dep. Energy* **2004**, 40–72.
2. Corsi, S.; Sabelli, C. General blackout in Italy Sunday September 28, 2003, h. 03:28:00. In *Proceedings of the 2004 IEEE Power Engineering Society General Meeting*; IEEE: Denver, CO, USA, 2004; pp. 1691–1702.
3. Chen, L.; Sun, Y.; Chen, X. Analysis of the Blackout in Europe on November 4, 2006. In *Proceedings of the 2007 IEEE Power Engineering Society General Meeting*, Tampa, FL, USA, 24–28 June 2007.
4. Dong, Z.; Xu, Y.; Wong, K.; Wong, K. Using IS to Assess an Electric Power System's Real-Time Stability. *IEEE Intell. Syst.* **2013**, 28, 60–66. [\[CrossRef\]](#)
5. Savulescu, S.C. *Real Time Stability Assessment in Modern Power Systems Control Centers*; Savulescu, S.C., Ed.; IEEE Press Series on Power Engineering; John Wiley & Sons: Piscataway, NJ, USA, 2009; ISBN 978-0-470-23330-6.
6. CIGRE Working Group. *601 CIGRE Technical Brochure on Review of On Line Dynamic Security Assessment Tools and Techniques*; CIGRE: Beijing, China, 2007.
7. Kundur, P.; Cañizares, C.; Paserba, J.; Ajarapu, V.; Anderson, G.; Bose, A.; Haziargyriou, N.; Hill, D.; Stankovic, A.; Taylor, C.; et al. Definition and Classification of Power System Stability. *IEEE Trans. Power Syst.* **2004**, 19, 1387–1401.
8. Cañizares, C.A. Voltage Stability Assessment: Concepts, Practices and Tools, IEEE/PES Power System Stability Subcommittee. *Tech. Rep. SP101PSS* **2002**, 1, 1–38.
9. Gómez-Expósito, A.; Conejo, A.J.; Cañizares, C. *Electric Energy Systems: Analysis and Operation*, 1st ed.; CRC Press: New York, NY, USA, 2008; ISBN 0-8493-7365-4.
10. Torres, S.P.; Peralta, W.H.; Castro, C.A. Power system loading margin estimation using a neuro-fuzzy approach. *IEEE Trans. Power Syst.* **2007**, 22, 1955–1964. [\[CrossRef\]](#)
11. Haziargyriou, N.D.; Van Cutsem, T. Indices Predicting Voltage Collapse including Dynamic Phenomena. In *Proceedings of the CIGRE 1994*, Paris, France, 8 August–8 September 1994.
12. Morison, K. On-line dynamic security assessment using intelligent systems. In *Proceedings of the 2006 IEEE Power Engineering Society General Meeting*, Montreal, QC, Canada, 18–22 June 2006; p. 5.
13. Shaikh, F.A.; Asghar, J. Computational Intelligence and Voltage Stability Analysis for Mitigation of Blackout. *Int. J. Comput. Appl.* **2011**, 16, 2. [\[CrossRef\]](#)
14. Jeyasurya, B. Artificial neural networks for on-line voltage stability assessment. In *Proceedings of the 2000 IEEE Power Engineering Society Summer Meeting*, Seattle, WA, USA, 16–20 July 2000; Volume 4, pp. 2014–2018.
15. Chakrabarti, S.; Jeyasurya, B. On-line voltage stability monitoring using artificial neural network. In *Proceedings of the 2004 Large Engineering Systems Conference on Power Engineering*, Halifax, NS, Canada, 28–30 July 2004; pp. 71–75.
16. Nakawiro, W.; Erlich, I. Online voltage stability monitoring using Artificial Neural Network. In *Proceedings of the Third International Conference on Electric Utility Deregulation and Restructuring and Power Technologies—DRPT 2008*, Nanjing, China, 6–9 April 2008; pp. 941–947.
17. Zhou, D.Q.; Annakkage, U.D.; Rajapakse, A.D. Online Monitoring of Voltage Stability Margin Using an Artificial Neural Network. *IEEE Trans. Power Syst.* **2010**, 25, 1566–1574. [\[CrossRef\]](#)
18. Goh, H.H.; Chua, Q.S.; Lee, S.W.; Kok, B.C.; Goh, K.C.; Teo, K.T.K. Evaluation for Voltage Stability Indices in Power System Using Artificial Neural Network. *Procedia Eng.* **2015**, 118, 1127–1136. [\[CrossRef\]](#)
19. Bulac, C.; Triștiu, I.; Mandiș, A.; Toma, L. On-line power systems voltage stability monitoring using artificial neural networks. In *Proceedings of the 2015 9th International Symposium on Advanced Topics in Electrical Engineering (ATEE)*, Bucharest, Romania, 7–9 May 2015; pp. 622–625.
20. Zhukov, A.; Tomin, N.; Sidorov, D.; Panasetsky, D.; Spirayev, V. A hybrid artificial neural network for voltage security evaluation in a power system. In *Proceedings of the 2015 5th International Youth Conference on Energy (IYCE)*, Pisa, Italy, 27–30 May 2015; pp. 1–8.
21. Subramani, C.; Jimoh, A.A.; Kiran, S.H.; Dash, S.S. Artificial neural network based voltage stability analysis in power system. In *Proceedings of the 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT)*, Nagercoil, India, 18–19 March 2016; pp. 1–4.
22. Bahmanyar, A.R.; Karami, A. Power system voltage stability monitoring using artificial neural networks with a reduced set of inputs. *Int. J. Electr. Power Energy Syst.* **2014**, 58, 246–256. [\[CrossRef\]](#)

23. Hashemi, S.; Aghamohammadi, M.R. Wavelet based feature extraction of voltage profile for online voltage stability assessment using RBF neural network. *Int. J. Electr. Power Energy Syst.* **2013**, *49*, 86–94. [\[CrossRef\]](#)
24. Rahi, O.P.; Yadav, A.K.; Malik, H.; Azeem, A.; Kr, B. Power System Voltage Stability Assessment through Artificial Neural Network. *Procedia Eng.* **2012**, *30*, 53–60. [\[CrossRef\]](#)
25. Innah, H.; Hiyama, T. A real time PMU data and neural network approach to analyze voltage stability. In Proceedings of the IEEE 2011 International Conference on Advanced Power System Automation and Protection, Beijing, China, 16–20 October 2011; pp. 1263–1267.
26. Ashraf, S.M.; Gupta, A.; Choudhary, D.K.; Chakrabarti, S. Voltage stability monitoring of power systems using reduced network and artificial neural network. *Int. J. Electr. Power Energy Syst.* **2017**, *87*, 43–51. [\[CrossRef\]](#)
27. Suganyadevi, M.V.; Babulal, C.K. Support Vector Regression Model for the prediction of Loadability Margin of a Power System. *Appl. Soft Comput.* **2014**, *24*, 304–315. [\[CrossRef\]](#)
28. Nizam, M.; Mohamed, A.; Hussain, A. Dynamic voltage collapse prediction in power systems using support vector regression. *Expert Syst. Appl.* **2010**, *37*, 3730–3736. [\[CrossRef\]](#)
29. Duraipandy, P.; Devaraj, D. On-line voltage stability assessment using least squares support vector machine with reduced input features. In Proceedings of the 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), Calcutta, India, 31 January–2 February 2014; pp. 1070–1074.
30. AbAziz, N.F.; Rahman, T.K.A.; Zakaria, Z. Voltage stability prediction by using Artificial Immune Least Square Support Vector Machines (AILSVM). In Proceedings of the 2014 IEEE 8th International Power Engineering and Optimization Conference (PEOCO), Langkawi, Malaysia, 24–25 March 2014; pp. 613–618.
31. Sajan, K.S.; Kumar, V.; Tyagi, B. Genetic algorithm based support vector machine for on-line voltage stability monitoring. *Int. J. Electr. Power Energy Syst.* **2015**, *73*, 200–208. [\[CrossRef\]](#)
32. Naganathan, G.S.; Babulal, C.K. Optimization of support vector machine parameters for voltage stability margin assessment in the deregulated power system. *Soft Comput.* **2018**, *23*, 10495–10507. [\[CrossRef\]](#)
33. Amroune, M.; Bouktir, T.; Musirin, I. Power System Voltage Stability Assessment Using a Hybrid Approach Combining Dragonfly Optimization Algorithm and Support Vector Regression. *Arab. J. Sci. Eng.* **2018**, *43*, 3023–3036. [\[CrossRef\]](#)
34. Huang, G.-B.; Zhou, H.; Ding, X.; Zhang, R. Extreme Learning Machine for Regression and Multiclass Classification. *IEEE Trans. Syst. Man Cybern. Part. B Cybern.* **2012**, *42*, 513–529. [\[CrossRef\]](#)
35. Duraipandy, P.; Devaraj, D. Extreme Learning Machine Approach for On-Line Voltage Stability Assessment. In *Proceedings of the Swarm, Evolutionary, and Memetic Computing*; Panigrahi, B.K., Suganthan, P.N., Das, S., Dash, S.S., Eds.; Springer International Publishing: Basel, Switzerland, 2013; pp. 397–405.
36. Suganyadevi, M.V.; Babulal, C.K. Online Voltage Stability Assessment of Power System by Comparing Voltage Stability Indices and Extreme Learning Machine. In *Proceedings of the Swarm, Evolutionary, and Memetic Computing*; Panigrahi, B.K., Suganthan, P.N., Das, S., Dash, S.S., Eds.; Springer International Publishing: Basel, Switzerland, 2013; pp. 710–724.
37. Duraipandy, P.; Devaraj, D. Extreme Learning Machine Approach for Real Time Voltage Stability Monitoring in a Smart Grid System using Synchronized Phasor Measurements. *J. Electr. Eng. Technol.* **2016**, *11*, 1527–1534. [\[CrossRef\]](#)
38. Zhang, R.; Xu, Y.; Dong, Z.Y.; Zhang, P.; Wong, K.P. Voltage stability margin prediction by ensemble based extreme learning machine. In Proceedings of the 2013 IEEE Power Energy Society General Meeting, Vancouver, BC, USA, 21–25 July 2013; pp. 1–5.
39. Ajarapu, V. *Computational Techniques for Voltage Stability Assessment and Control*; Springer: Berlin, Germany, 2006.
40. Lee, K.Y. Current Trend and the State of the Art in Intelligent System Applications to Power Systems. In Proceedings of the International Conference on Intelligent Systems Application to Power Systems (ISAP), Rio de Janeiro, Brazil, 4–8 April 1999; pp. 1–80.
41. Bishop, C.M. *Pattern Recognition and Machine Learning*; Springer: Berlin, Germany, 2006; Volume 4, ISBN 9780387310732.
42. Han, J.; Kamber, M.; Pei, J. *Data Mining: Concepts and Techniques*, 3rd ed.; Morgan Kaufmann: San Francisco, CA, USA, 2011; ISBN 0-12-381479-0.

43. Cepeda, J.C. Evaluación de la Vulnerabilidad del Sistema Eléctrico de Potencia en Tiempo Real usando Tecnología de Medición Sincrofasorial. Ph.D. Thesis, Universidad Nacional de San Juan, San Juan, Argentina, 2013.
44. Dong, Z.; Zhang, P.; Ma, J.; Zhao, J.; Ali, M.; Meng, K.; Yin, X. *Emerging Techniques in Power System Analysis*; Springer Science & Business Media: Berlin, Germany, 2010; ISBN 978-3-642-04282-9.
45. Abe, S. *Support. Vector Machines for Pattern Classification*; Springer Science & Business Media: Berlin, Germany, 2006; ISBN 978-1-84628-219-5.
46. Hsu, C.; Chang, C.; Lin, C. *A Practical Guide to Support Vector Classification*; Department of Computer Science, National Taiwan University: Taipei, Taiwan, 2010.
47. Chang, C.-C.; Lin, C.-J. LIBSVM: A Library for Support Vector Machines. *ACM Trans. Intell. Syst. Technol.* **2011**, *2*, 27. [\[CrossRef\]](#)
48. Huang, G.-B.; Zhu, Q.-Y.; Siew, C.-K. Extreme learning machine: Theory and applications. *Neurocomputing* **2006**, *70*, 489–501. [\[CrossRef\]](#)
49. Huang, G.-B.; Wang, D.H.; Lan, Y. Extreme learning machines: A survey. *Int. J. Mach. Learn. Cybern.* **2011**, *2*, 107–122. [\[CrossRef\]](#)
50. Huang, G.-B. An Insight into Extreme Learning Machines: Random Neurons, Random Features and Kernels. *Cogn. Comput.* **2014**, *6*, 376–390. [\[CrossRef\]](#)
51. Tu, J.; Xu, Y.; Yin, Z. Data-Driven Kernel Extreme Learning Machine Method for the Location and Capacity Planning of Distributed Generation. *Energies* **2019**, *12*, 109. [\[CrossRef\]](#)
52. Billinton, R. *Reliability Evaluation of Power Systems*; Springer Science & Business Media: Berlin, Germany, 2013; ISBN 978-1-4615-7731-7.
53. Wang, X.-F.; Song, Y.-H.; Irving, M. *Modern Power Systems Analysis*; Springer: Berlin, Germany, 2008; ISBN 978-0-387-72853-7.
54. Cañizares, C.A.; Alvarado, F.L. Point of collapse and continuation methods for large AC/DC systems. *IEEE Trans. Power Syst.* **1993**, *8*, 1–8. [\[CrossRef\]](#)
55. Ajarapu, V.; Christy, C. The continuation power flow: A tool for steady state voltage stability analysis. *IEEE Trans. Power Syst.* **1992**, *7*, 416–423. [\[CrossRef\]](#)
56. Sundaram, K.; Swarup, S. Classification and Assessment of Power System Security Using Multiclass SVM. *IEEE Trans. Syst. Man Cybern. Part. C Appl. Rev.* **2011**, *41*, 753–758.
57. Erlich, I. Mean-Variance Mapping Optimization Algorithm Home Page. Available online: <https://www.uni-due.de/mvmo/> (accessed on 9 February 2018).
58. Rueda, J.L.; Cepeda, J.C.; Erlich, I. Estimation of location and coordinated tuning of PSS based on mean-variance mapping optimization. In Proceedings of the 2012 IEEE Power and Energy Society General Meeting, San Diego, CA, USA, 22–26 July 2012; pp. 1–8.
59. Rueda, J.L.; Erlich, I. Optimal dispatch of reactive power sources by using MVMOs optimization. In Proceedings of the 2013 IEEE Computational Intelligence Applications in Smart Grid (CIASG), Singapore, 16–19 April 2013; pp. 29–36.
60. Pai, M.A. *Energy Function Analysis for Power System Stability*; Springer: Berlin, Germany, 1989; ISBN 978-0-7923-9035-0.
61. Washington, U. Power Systems Test Case Archive—UWEE. Available online: <http://www2.ee.washington.edu/research/pstca/> (accessed on 16 April 2018).
62. Huang, G.-B. Extreme Learning Machines: Random Neurons, Random Features, Kernels. Available online: <http://www.ntu.edu.sg/home/egbhuang/> (accessed on 23 April 2018).
63. Math Works, Inc. MATLAB. Available online: <https://la.mathworks.com/products/matlab.html> (accessed on 24 April 2018).
64. Glavic, M.; Van Cutsem, T. Wide-Area Detection of Voltage Instability from Synchronized Phasor Measurements. Part II: Simulation Results. *IEEE Trans. Power Syst.* **2009**, *24*, 1417–1425. [\[CrossRef\]](#)

