Real-Time Monitoring of Long-Term Voltage Stability via Convolutional Neural Network

Shiyang Li
Department of Electrical and Computer Engineering
Iowa State University, Ames, Iowa, USA
Email: shiyangl@iastate.edu

Venkataramana Ajjarapu
Department of Electrical and Computer Engineering
Iowa State University, Ames, Iowa, USA
Email: vajjarap@iastate.edu

***Abstract*—Convolutional Neural Network (CNN) is one of the most promising deep learning technique that has achieved great success in many areas. In this paper, we implement a CNN for real-time monitoring of long-term voltage stability margin (VSM). To clarify our motivations of introducing CNN in this problem, we first discuss the essence and the complicity of VSM prediction, and summarize the limitations of existing methods, then point out the input structure of CNN inherently contains the topology information of power network, thus could have great potential in solving the problem. An CNN architecture, together with an input encoding method that strengthens topology information, is proposed and tested on IEEE 30-bus system. Preliminary results show that it can achieve better prediction performance comparing to some existing methods, and can be successfully employed in online voltage stability monitoring.**

I. Introduction

Voltage collapse played important roles in many blackout incidents, including the northeast blackout in 2003 [1]. The countermeasures include evaluation and adoption of better real-time tools for operators and reliability coordinators. This stimulates a continually increasing interest in online voltage stability assessment. One major category of online voltage stability assessment is monitoring the voltage stability margin (VSM) based on the wide area measurements. However, the closed form solution of VSM for a real system does not exist. So calculating the exact value of VSM is usually computationally intensive and impractical for online applications.

Machine learning techniques enabled VSM online monitoring by transferring the computationally intensive simulation process from online to off-line. Predictive models are trained using the data from off-line simulation, and then used online to predict VSM. Various learning techniques have been investigated for this purpose, such as artificial neural network (NN) [2], decision tree (DT) [3], regression tree (RT) [4], bagging [5], linear regression [6] and local regression [7].

In these works, due to the model flexibility, NN-based methods usually achieved better off-line prediction accuracy. However, there exists some inherent limitations. [5] argued that the opacity of NN impairs its applicability in security-sensitive task like power system stability monitoring and control. [8] tried to solve this problem by whitening the NN black box using DT-based rule revealing. Another criticism is that the NN must be trained separately for different system topology in many early work. But [9] showed the possibility to get a single NN that is universal from a bunch of topologies. However, the effective capacity of shallow NN with respect to the complicated power system operation space is not discussed, and only one- or two-layer NN has been implemented to the best of our knowledge. To balance the simplicity/transparency and the accuracy, [7] proposed to use adaptive local linear regression to circumvent the model genericity issue. However, the local linear model, agile but weak, will be haunted by the genericity issue during the adapting processes for an unforeseen operating condition, and consequently fail to provide trustable prediction when the operator care it the most.

Convolutional Neural Network (CNN) [10], is a specialized kind of neural network for processing data that has a known, grid-like topology. It has been tremendously successful in practical applications, such as image pattern recognition, and board game AI including the stirring victory of AlphaGo. The special structure of CNN liberates the power of deep NN in practice, especially for the sophisticated tasks that a simple leaner usually cannot help. However, little work can be found in power system area that utilizes this powerful tool inherently connecting to a network-like object.

This paper introduces CNN to solve the online VSM prediction problem. Benefiting from the deep architecture and the effective input structure of CNN, the genericity of VSM predictive model is expected to be extend to a new level.

The rest of the paper is organized as follows: Section II present the motivation of applying CNN in VSM prediction; Section III described our implementation of CNN; test results are presented in Section IV, followed by the conclusions in Section V, presenting the major findings and future work.

II. Opportunity of Employing CNN in VSM Prediction

A. Definitions and Terminology

To facilitate our discussion, first we declare some general terms. The *voltage stability margin (VSM)*, denoted by M, refers to the distance between current operating point and the *critical point* (nose point of PV curve) measured by the total real power load increment. *Reactive power reserve (RPR)* of a reactive power source j is defined by the difference between the maximal reactive generation Q_j^{max} and the current reactive generation Q_j. Except RPR, following the suggestion of [6], voltage magnitudes and active power flows are also

978-1-5386-2212-4/17/$31.00 ©2017 IEEE

used as VSM predictors. We use $X = (RPR, V, P_{flow})$, where $X \in \mathbb{R}^{1\times D}$ and D is the total number of selected RPRs, voltage magnitudes and active power flows, to denote the input variables for VSM perdition.

Tracing the PV curve by continuation power flow or other methods gives a series of operating points, and then yields the corresponding (X, M) pairs. For a comprehensive long-term *voltage stability assessment (VSA)*, PV curves are traced off-line under various *contingencies* (in this paper, outages of generators, transformers and transmission lines are considered), *operating scenarios* (certain system configurations such as generation and load profile, control parameters) and *load increase directions (LIDs)*. This builds up a *database* of (X, M) pairs.

B. The Complexity of VSM Prediction Problem

In II-A, we have described how to calculate VSMs under various operating conditions through the procedure of VSA (set parameters of operating condition, then trace PV curve). Our objective is to find the predictive model that explicitly maps online measurements (reflecting the operating condition) to VSM, capturing their relationship that implicitly exists in the procedure of VSA. In this part, we will investigate the abstract equations defining that underlying relationship, i.e., the mathematical equivalent to the VSA procedure. Then we can find that the underlying relationship determined by these equations, i.e., the *learning target* of the predictive model, is too complex to be represented by a practical-sized shallow model, thus a deep model is more suitable.

Under reasonable assumptions, the critical point of voltage stability is connected to the saddle-node bifurcation (SNB), which can be described by (ignore transversality conditions)

$$f(x, \lambda, p) = 0 \tag{1}$$

$$g(x, \lambda, p) = \det(f_x) = 0 \tag{2}$$

f represents the static system equations (usually power flow equations). x is the state variable. λ is the load increment (at a SNB) with respect to certain base value. p is any continuous parameter giving the operating condition, such as the base load profile and the LID. Equation (2) (or its variance) is called the bifurcation equation, saying the Jacobian $\partial f/\partial x$ is singular. [11] shows that locally there exists a smooth implicit function h that maps p to λ. If h is the learning target, then by the universal approximation theorem, it is possible to accurately represent it by a shallow NN.

Unfortunately, (1)-(2) are not equivalent to the VSA procedure, so h is not our learning target. First, they are valid for all SNBs. But the only SNB making physical sense, which makes λ equal to M obtained by PV curve tracing, is the one connecting stable equilibrium manifold. So, if multiple SNBs exist for fixed p (generally true), extra information is needed to determine the h with physical sense.

Second, (1)-(2) cannot describe the discrete changes of operating condition in VSA such as bus type switching (e.g. PV→PQ), tap changing, shunt switching, unit start-up/shut-down, and contingencies. These changes are usually regarded as the changes of f and g, so cannot be described universally by a single set of equations. To describe the learning target valid for all operating conditions, we need the equations

$$\bar{f}_1(x, M, Y_{bus}(p, d), p, d) = 0 \tag{3}$$

$$\bar{f}_2(x, M, Y_{bus}(p, d), p, d) = 0 \tag{4}$$

$$\bar{g}(x, M, Y_{bus}(p, d), p, d) = 0 \tag{5}$$

$$\bar{\rho}(x, M, Y_{bus}(p, d), p, d) = 0 \tag{6}$$

Y_{bus} is the network admittance matrix (notice that most of the discrete changes can be represented by the variation of Y_{bus}). d is any discrete parameter (e.g. on-off states of units, tap positions). $\bar{f}_1$ is the full power balance equations. $\bar{f}_2$ describes the conditional constraints, typically the ones control the bus types (e.g. $(Q_i - Q_i^{max})\, d_k = 0$, where d_k is an element of d such that d_k is nonzero iff Q_i reaches its upper limit). Equation (5) represents the bifurcation equation. Equation (6) is the condition that restricts the SNB to be the one with physical sense (could be realized by the holomorphic embedding method [12]). Thus, the mapping $\bar{h} : (p, d) \mapsto M$, if uniquely determined by (3)-(6), explicitly links the parameters to VSM on quite general operating conditions.

However, we do not usually regard $\bar{h}$ as the learning target because p and d are unsuitable to be directly used as the predictors of a predictive model. In particular, (i) $\dim p + \dim d$ (the total dimension of parameters) is usually very large; (ii) parts of p and d are unobservable from EMS; (iii) parts of p and d are inefficient as VSM predictors. Instead, we select some online measurements

$$y = \mathfrak{O}(x, p, d) \tag{7}$$

to be the predictors (RPR, V, and P_{flow} in this paper). Since $\dim y \ll \dim p + \dim d$, $\mathfrak{O}^{-1}$ generally does not exist. Thus, $\tilde{h} : y \mapsto M$ is undetermined by (3)-(7). But if (p, d) has certain joint probability distribution $\mathscr{P}$ for practical operating conditions, then the conditional expectation $\mathbb{E}(p, d \,|\, y)$ can be obtained via (3)-(7), then there exists $\mathring{h} : y \mapsto \mathbb{E}(M)$ determined by (3)-(7) and $\mathscr{P}$.

This $\mathring{h}$, if exists, is our learning target. Unfortunately, each of (3)-(7) or $\mathscr{P}$ is highly nonlinear, discrete and more importantly, full of composite logics (which is a critical prior of deep model). As a result, $\mathring{h}$, which gives the expectation of VSM on any practical operating condition, is very complex and difficult to be accurately learnt by a practical-sized NN or other shallow models that have limited effective capacity. Many evidences show that this kind of complex learning target is more suitable to be treated by deep learning methods [13].

C. The Limitation of Existing Methods

II-B explained why most of the existing methods have to train a bunch of predictive models to deal with varying operating conditions, especially different topologies. The consequences of having multiple model include: (i) for the method that links each model explicitly to certain operating conditions (e.g. [5]), we may loss the genericity for the operating conditions outside the training set (prediction degenerates to looking

978-1-5386-2212-4/17/$31.00 ©2017 IEEE

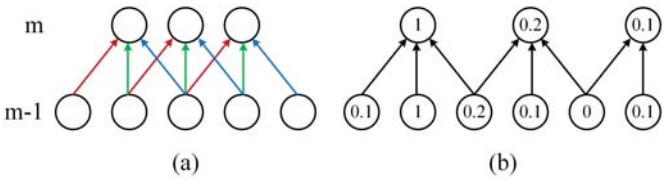

Fig. 1. Conceptual plots for the key components of a CNN. (a) The convolution layer. The arrows with same color indicate the same weight. (b) The maxPooling layer. The output of the maxPooling on layer m is almost invariant if the inputs from layer m-1 move to right or left by one neuron.

up table); (ii) for the method that relies on an extra learner to select the effective model for given operating condition (e.g. [6]), the extra learner itself will face the capacity limit, and the genericity of a few predictive models is also limited.

Thus, to overcome these limitations, reasonably we should either (i) give up the genericity of off-line predictive model and train the relatively low-capacity but adaptive model online to fit the new operating condition, or (ii) enhance the off-line predictive model with more powerful structure, namely the deep network. Our previous work [7] make efforts on (i), and this paper will advance on the direction of (ii).

D. The Structure of CNN

A CNN emulates the behavior of animal visual cortex by a multilayer NN equipped with two special kinds of layers. The first one is called the *convolution layer*, which is characterized by sparse connectivity and weights sharing (Fig.1(a)). If we regard the shared weights as a filter that take a small region of layer m-1 as input, layer m is just the overlapping of the output of the filter when it sweeps the whole layer m-1. Mathematically, this is the convolution of layer m-1 and the weights vector. The major benefits of this structure include: (i) the number of weights is significantly reduced, which is critical for the success of of a deep network; (ii) the pattern that recognized by the filter is translation invariant.

The other special type of layer is the pooling layer (typically the max-pooling, see Fig.1(b)). Max-pooling partitions the input image into rectangles and, for each rectangle, outputs the maximum value. Max-pooling is useful for two reasons: (i) the down-sampling process reduces computation for upper layers; and (ii) it provides additional robustness to position (and shape) perturbations because the maximum function ignore the movement of pixel within the rectangles.

E. The Favorable nature of CNN for VSM Prediction

The success of CNN in complex pattern recognition, where the relationship between input and output is also highly non-linear and discrete, inspires us to apply it to VSM prediction problem. Besides, Some inherent features of VSM prediction problem that could make CNN particularly suitable. For example, power network can be seen as a two-dimensional object, namely, a graph. So there inherently exists topology structure among the dimensions of input space. This topology structure is actually a cornerstone of the underlying pattern through Y_{bus} in (3), and therefore it is evidently useful in VSM prediction. This network topology information can be naturally carried by an second-order tensor (an image, further explained in III-A) as the input to a CNN. In contrast, all existing learning-based VSM prediction methods use a randomly ordered first-order tensor (vector) to represent the inputs, which loss the valuable topology information. Predicting VSM via CNN emulates that a mature operator in control center usually can intuitively percept the system security risk by a glance at the power flow monitoring screen, which is more efficient than reading lots of numbers in tables.

III. Applying CNN in VSM Prediction

A. Inputs Encoding

To prepare the input for a CNN, we need to transfer the predictors (online measurements) RPR, V, and P_{flow} of an operating point into an image. A nature design would be: (i) represent each class of variables by a separate image channel (so here we have three channels, i.e. RGB, to represent RPR, V, and P_{flow} respectively); (ii), the value of each variable (after bias and scaling) will be represented by the color of a pixel; (iii) arrange the variable positions in the form of incidence matrix, which means, RPR_i and V_i, the RPR and voltage magnitude at bus i, should be represented by the pixel at the diagonal position (i, i), and P_{ij}, the active power from bus i to bus j, should be represented by the pixel at (i, j). Thus, the inputs of an n-bus system will be represented by an n-by-n colorful image. Notice that the image can be store as sparse matrix. So, nearly no extra space is imposed for the image representation. Fig.2 shows four examples. Notice that along the diagonal, brighter red indicates larger RPR; brighter green indicates higher voltage; in the off-diagonal area, brighter blue indicates larger flow in the positive direction (dark blue represent the negative flow, so zero flow is shown in medium blue).

Furthermore, the order or buses can be optimized to strengthen the topology information. The target is to make two buses with small electrical distance (large connecting admittance) to be also close in terms of bus index ($|i - j|$ for bus i and j). In this way, more topology information could be represented. For example, the low voltage of certain area will be reflected by the dark area at the diagonal, which could be more visually recognizable than a bunch of separate dark spots. This reordering problem can be formulated as

$$\begin{aligned} \min_{P} \quad & \langle P^T A P \, , \, W \rangle \\ & a_{ij} = |i - j| \\ & w_{ij} = |Y_{ij}| \\ \text{s.t.} \quad & \forall i, j : p_{ij} \in \{0, 1\} \, , \, \sum_{i=1}^{n} p_{ij} = 1 \, , \, \sum_{j=1}^{n} p_{ij} = 1 \end{aligned} \tag{8}$$

where P, A, and W are n-by-n matrixes (for n-bus system); a_{ij}, w_{ij} and p_{ij} are their entries respectively. A is the matrix of index distance; W is the matrix of reciprocal electrical distance, where Y_{ij} is the entry of Y_{bus}; the last constrain restricts P to be a permutation matrix. $\langle \, , \, \rangle$ is the inner product operator of matrixes. Thus, the objective is the sum of (index distance

978-1-5386-2212-4/17/$31.00 ©2017 IEEE

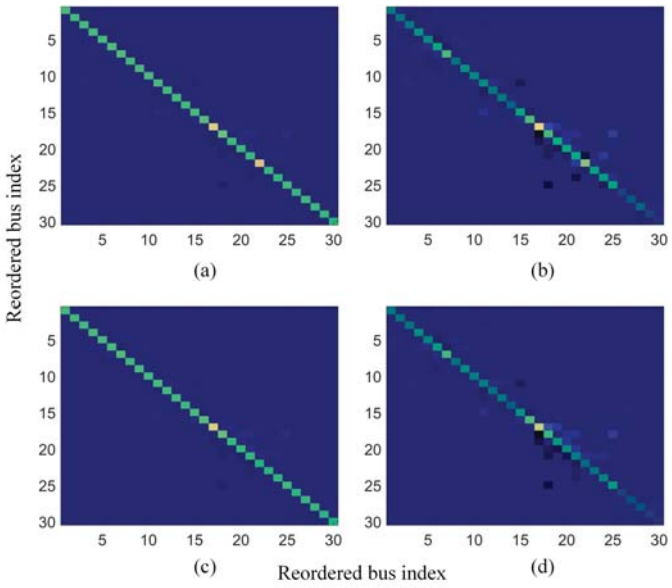

Fig. 2. The image representations of inputs for IEEE 30-bus system. (a) Base point under normal condition. (b) Critical point under normal condition. (c) Base point when generator at bus 8 (becomes bus 22 after reordering) is tripped. (d) Critical point when generator at bus 8 is tripped.

after reordering $\times$ the reciprocal of electrical distance) for all bus pairs. This is a typical quadratic assignment programming, which is a hard NP problem without efficient analytical algorithm for the general case. However, the global optimality is not necessary for our purpose, acceptable solutions can be obtained by heuristic algorithms (Genetic in this paper). Encoding the inputs into an image can be regarded as a feature mapping enriched with the topology information of the system.

B. The Architecture of The CNN

We utilize the CNN models of MATLAB. The architecture of the implemented CNN is shown in Fig.3. The ReLU (Rectified Linear Unit) layer performs a threshold operation, which is the standard setting of activation function for deep network because the threshold increases the sparsity of the network and the constant slop reduces the likelihood that the gradient quickly decays during backpropagation. The dropout layer randomly drops out some neurons on the previous layer during the training iteration. It reduces coadaption between neurons, so as to prevent overfitting. Originally designed for classification, the CNN model ends up with a fully connected layer, a softmax layer and a classification layer. We deploy maxPooling layers every two convolution layers to avoid the improperly large down-sampling ratio for small input images.

C. Categorization and Smoothing of VSM

To fit the output structure of the CNN, the VSM range of training set is equally partitioned into sections (the ith category covers VSM from $(i-1)\,r$ to ir MW, where r is the section range; r = 10 MW in the test example). Thus, VSM prediction is temperately treated as a classification problem.

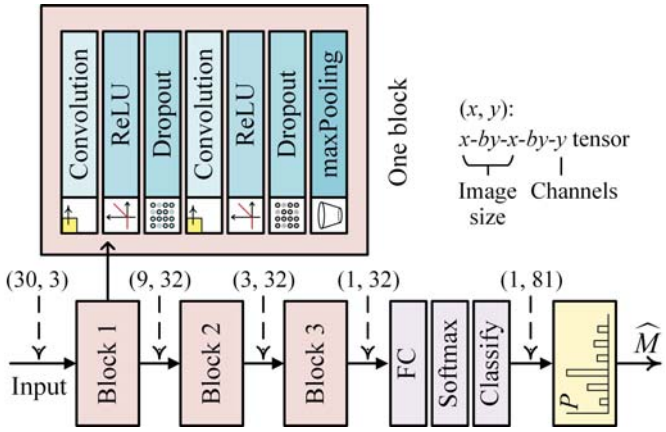

Fig. 3. The architecture of the implemented CNN for IEEE 30-bus system. 6 convolution, 3 maxPooling layers and 1 fullConnected layer are included. The size of data between the blocks are annotated.

For any given input, the CNN can output the probabilities that the VSM belongs to each category. Then we can smooth the VSM prediction by

$$\widehat{M} = \sum_{i=1}^{m} p_i^s \,(i-1/2)\, r \Big/ \sum_{i=1}^{m} p_i^s \tag{9}$$

where p_i is the predicted probability that M belongs to the ith category; s is some positive number than can be tuned during the validation process; $s = 4$ in the test example.

D. The Training Algorithm

The proposed CNN is trained via standard mini-batch stochastic gradient descent algorithm. The algorithm parameters (learning rate, match size, L_2 regularization factor, momentum, etc.) and the architecture parameters (number and size of filters, etc.) are mainly manually tuned base on experience rules [14].

IV. Example

IEEE 30-bus system is used to test the proposed approach.

A. The Database

A summary of the database is shown in Table I. The division of database and the distribution of contingencies and LIDs are intentionally designed to challenge the genericity of the predictive model for unforeseen operating condition, which is expected to be an advantage of the proposed approach comparing to existing ones.

B. Prediction Performance

A CNN with 32 filters for each convolution layer is implemented. Besides, three other approaches are implemented for comparison. There methods are based on our previous work so it is possible to fairly compare them on the uniform base. The prediction root-mean-square-errors (RMSE) are shown in Table.II. It shows that the proposed CNN achieve the best performance on the test set. More importantly, the error on the test set and that on the validation set are quite close, which

978-1-5386-2212-4/17/$31.00 ©2017 IEEE

TABLE I
SUMMARY OF THE DATABASE

System	Num_Con				Num _LID	Size	Num _Mes
	N-1	N-2	N-3	N-4			
IEEE30	43	250	250	146	40	329226	73

Num_con: counts of contingencies. Num_LID: counts of LIDs. Size: counts of operating points in the database. Num_Mes: counts of monitored online measurements, including RPRs, voltage magnitudes and real power flows.

N-1 contingencies include outages of all non-slack generators, transformers, and transmission lines. N-2, N-3, and N-4 contingencies are random combinations of N-1 contingencies. LIDs are randomly sampled from Gaussian distribution with the base load profile as the expectation. The database is divided into the training set (40%), validation set (30%) and the test set (30%). 25% of the N-2, N-3, N-4 contingencies, LIDs are preserved for validation and test set respectively. E.g. 10 LIDs in validation set are never used in training set; 10 LIDs in test set are never used in training and validation set. So, the contingencies, operating scenarios and LIDs in validation sets and test sets are roughly half known and half unforeseen.

PSS/E is used for VSA. ± 50 MW perturbations on re-dispatching of generations and loads, and ± 0.015 p.u. perturbations on generator scheduled voltage are applied to the base point of each PV curve.

TABLE II
PREDICTION PERFORMANCE IN RMSE (UNIT: $\text{VSM}_{\text{initial}}\%$)

MLRM			LR		NN		CNN		
Tr	V	Ts	V	Ts	Tr	Ts	Tr	V	Ts
4.55	5.22	11.91	5.34	6.30	5.86	6.70	2.70	4.29	4.30

$\text{VSM}_{\text{initial}}$ = 673.67 MW, which is the average VSM on normal operating condition in training set.

MLRM: multilinear regression models, based on [6], trained using LASSO. Three quadratic MLRMs are established, and the error of model identification tool is ignored for simplification. LR: local linear regression with kernel weights, based on [7], trained using LASSO. NN: a regular feedforward neural network with two hidden layers, based on Neural Network Fitting Tool of MATLAB, trained by scaled conjugate gradient (SCG) backpropagation [15] (achieves best result among several algorithms). We choose 2 as the number of NN hidden layers because far as we know, the existing NN-based VSM prediction methods rarely use more than 2 hidden layers.

Tr: training set; V: validation set; Ts: test set. LR is nonparametric that does not have a training process. SCG uses regularization rather than validation set to tune the hyperparameters.

In this example MLRM gives a small amount of extremely bad predictions (error $>$ 1000 MW). If they are regarded as outliers and deleted, the RMSE for MLRM is 10.04%.

implies the CNN has a good genericity. Besides, we also tried a 10-layer regular NN. 10 is the total number of convolution, maxPolling, and fullConnected layers, so the CNN is a special case of this network. When the depth goes to this level, regular NN becomes very sensitive to the stochastic factors of the training process, and the errors of the training set and the test set diverges significantly in our observation. The best result we got for the 10-layer NN is 6.87%, even worse than a two-layer one. In contrast, benefiting from the smart prior involved in CNN, overfitting was restrained fairly well. In fact, parameters of the CNN here are tuned manually, which implies that its performance could be further enhanced if an effective hyperparameter tuning scheme is applied.

V. CONCLUSION

Based on the natures and the connection between VSM prediction problem and convolutional neural network, an on-line voltage stability margin monitoring approach using CNN is proposed. Selected online measurements such as RPRs, voltage magnitudes and active power flows, are encoded into a multi-channel image according to their topological relationship in the power network. This image is then fed into the CNN as inputs. Separate data-sets, which are generated from off-line VSA and have distinct contingencies and LIDs, are used to train, validate and test the CNN with 10 major hidden layers. Preliminary results shows that the proposed approach achieves better prediction accuracy comparing to several reference methods, which extends the boundary of predictive model generality for VSM prediction problem.

To compare the proposed approach with others on a uniform base, the input online measurements are restricted to the three quantities mentioned above. So, the input image is quite sparse. In other words, there are a lot of free "slots" in the input framework ready for more measurements. For example, the off-line region of RPR channel can be utilized to include reactive line flows. This further implies the potential of the proposed approach. Utilizing more quantities and scaling up the CNN for large system will be investigated in future.

REFERENCES

[1] "Final report on the august 14, 2003 blackout in the united states and canada: causes and recommendations," U.S.-Canada Power System Outage Task Force, Tech. Rep., April 2014.

[2] A. El-Keib and X. Ma, "Application of artificial neural networks in voltage stability assessment," *IEEE Trans. Power Syst.*, vol. 10, no. 4, pp. 1890–1896, 1995.

[3] T. Van Cutsem, L. Wehenkel, M. Pavella, B. Heilbronn, and M. Goubin, "Decision tree approaches to voltage security assessment," *IEE Proceedings C Generation, Transmission and Distribution*, vol. 140, no. 3, pp. 189–198, 1993.

[4] C. Zheng, V. Malbasa, and M. Kezunovic, "Regression tree for stability margin prediction using synchrophasor measurements," *IEEE Trans. Power Syst.*, vol. 28, no. 2, pp. 1978–1987, 2013.

[5] Y. Fan, S. Liu, L. Qin, H. Li, and H. Qiu, "A novel online estimation scheme for static voltage stability margin based on relationships exploration in a large data set," *IEEE Trans. Power Syst.*, vol. 30, no. 3, pp. 1380–1393, May 2015.

[6] B. Leonardi and V. Ajjarapu, "Development of multilinear regression models for online voltage stability margin estimation," *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 374–383, 2011.

[7] S. Li and V. Ajjarapu, "Real-time monitoring of long-term voltage stability via local linear regression," in *2015 IEEE Power Energy Society General Meeting*, July 2015.

[8] R. Setiono, W. K. Leow, and J. Zurada, "Extraction of rules from artificial neural networks for nonlinear regression," *IEEE Trans. Neural Netw.*, vol. 13, no. 3, pp. 564–577, 2002.

[9] D. Zhou, U. Annakkage, and A. Rajapakse, "Online monitoring of voltage stability margin using an artificial neural network," *IEEE Trans. Power Syst.*, vol. 25, no. 3, pp. 1566–1574, 2010.

[10] Y. LeCun, "Generalization and network design strategies. technical report crg-tr-89-4," University of Toronto, Tech. Rep., 1989.

[11] S. Greene, I. Dobson, and F. L. Alvarado, "Sensitivity of the loading margin to voltage collapse with respect to arbitrary parameters," *IEEE Trans. Power Syst.*, vol. 12, no. 1, pp. 262–272, Feb. 1997.

[12] S. S. Baghsorkhi and S. P. Suetin, "Embedding ac power flow in the complex plane part i: Modelling and mathematical foundation," *arXiv preprint arXiv:1604.03425*, 2016.

[13] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT Press, 2016.

[14] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," in *Neural Networks: Tricks of the Trade*. Springer, 2012, pp. 437–478.

[15] M. F. Møller, "A scaled conjugate gradient algorithm for fast supervised learning," *Neural networks*, vol. 6, no. 4, pp. 525–533, 1993.

978-1-5386-2212-4/17/$31.00 ©2017 IEEE