Voltage Sag Estimation in Sparsely Monitored Power Systems Based on Deep Learning and System Area Mapping

DOI:
10.1109/TPWRD.2018.2865906

Document Version
Accepted author manuscript

Citation for published version (APA):
Liao, H., Milanovic, J. V., Rodrigues, M., & Shenfield, A. (2018). Voltage Sag Estimation in Sparsely Monitored Power Systems Based on Deep Learning and System Area Mapping. *IEEE Transactions on Power Delivery*, *33*(6), 3162-3172. https://doi.org/10.1109/TPWRD.2018.2865906

Published in:
IEEE Transactions on Power Delivery

Voltage Sag Estimation in Sparsely Monitored Power Systems Based on Deep Learning and System Area Mapping

Huilian Liao, *Member, IEEE*, Jovica V. Milanović, *Fellow, IEEE*, Marcos Rodrigues, Alex Shenfield

Abstract--**This paper proposes a voltage sag estimation approach based on a deep convolutional neural network. The proposed approach estimates the sag magnitude at unmonitored buses regardless of the system operating conditions and fault location and characteristics. The concept of system area mapping is also introduced via the use of bus matrix, which maps different patches in input matrix to various areas in the power system network. In this way, relevant features are extracted at various local areas in the power system and used in the analysis for higher level feature extraction, before feeding into a fully-connected multiple layer neural network for sag classification. The approach has been tested on the IEEE 68-bus test network and it has been demonstrated that the various sag categories can be identified accurately regardless of the operating condition under which the sags occur.**

Index Terms—**Voltage sag estimation, deep learning, convolutional neural networks, bus matrix, pattern classification.**

I. Introduction

Voltage sags, as one of the most critical power quality issues, continue to attract great attention from both utilities and industries, due to its frequent disruption of industrial processes, adverse impacts on electronic equipment and the resultant substantial financial losses [1]. Proper voltage sag estimation at buses of interest, be monitored or not, can be used to assist sag mitigation planning. Voltage sag performance of the network can be established reasonably accurately by a long term monitoring at sufficient number of locations. This however has proved to be costly and various methods have been developed to assess voltage sag performance based on a limited number of accessible meters [2].

Voltage sag estimation can be mainly classified into two categories, sag profile estimation and sag performance estimation [3]. Sag profile estimation is to estimate the voltages at unmonitored buses during a single fault event. Since sags are mainly caused by faults in power systems, estimation through fault location/identification is extensively explored for sag profile estimation [4]. In these approaches, fault type and location are identified first, followed by the estimation of sag profiles at unmonitored buses [5, 6]. As a model-based method, fault position uses circuit analysis to calculate sags during faults at specific locations. The estimation requires more detailed network information (including fault impedance) and the obtained results are highly dependent on the choice and number of fault positions [7]. In [8] the concept of 'fault position' is adopted and the estimation requires the prior information on network conditions including load and DG generation etc. The method of critical distances, as an alternative to fault positions, requires less data but is limited for application in radial systems only [9].

Instead of estimating detailed sag profiles as implemented in sag profile estimation, sag performance estimation is concerned with parameter estimation including the number of voltage sags or bins of voltage depths at unmonitored buses and/or of the system [7]. Voltage sag performance can be estimated using either statistical analysis of historical records or probabilistic assessment such as Monte Carlo simulation which is carried out based on given fault probabilities of various components in the network [10]. Statistical analysis can be implemented using classical state estimation formulation combined with historical record of measurements [11]. In [12], the sag occurrence frequency (SARFI indices) is estimated based on processing a database of voltage sag scenarios using Bayesian filtering method. In [13], the number of voltage sags occurring at unmonitored buses is derived from the number of sags recorded at monitored buses by constructing a measurement matrix, while in [14], a general neural network is used to estimate number of sags within two classes based on data obtained from a relatively simpler 24 bus test network. Analytical analysis based on detailed historical measurement records provides accurate assessment [15], however it highly relies on the availability of the sag records in this case.

Deep learning techniques, especially convolutional neural networks (CNNs), have been under development for a few decades [16]. By 2012 they attracted great attention due to impressive results achieved in a large scale visual recognition competition [17]. Deep learning techniques in general require multiple processing hierarchical layers and a large computational effort to deal with large amounts of data [18]. They have been found though to facilitate extraction of more

H. Liao is with Power, Electrical and Control Engineering Group (email: huilian.liao@shu.ac.uk), and M. Rodrigues and A. Shenfield are with Geometric Modelling and Pattern Recognition Research Group at Sheffield Hallam University, Sheffield S1 1WB, UK. J.V. Milanović is with the School of Electrical and Electronic Engineering, The University of Manchester, PO Box 88, Manchester, M60 1QD, UK.

informative features for machine learning applications [19] and typically outperform general neural networks in solving problems with multiple data classes and complex data structures. Thanks to recent development in high-performance hardware they become feasible and popular option in various applications. CNNs are one of the most popular deep learning methods used for image/pattern recognition. They consist of convolution layers followed by a fully-connected (FC) neural network. In convolution layers a set of learnable filters/kernels slide through input data to characterize the data. Multiple layers are typically used in both convolutional layers and neural network layers. The potential of applying deep learning in power system analysis has also been explored. Auto-encoder, an unsupervised deep learning approach, is applied for load profile classification [20]. CNNs have been applied to estimate the state-action value function in supervised learning for residential load control [21]. The application of CNNs at system level, however, is still limited. Considering their powerful pattern recognition capability their suitability for solving various recognition/estimation problems at power system level should be further explored.

This paper introduces the CNN based VSE approach to estimate voltage sag magnitudes with a high accuracy regardless of the uncertainties associated with load demand, DG outputs, fault types, fault area and fault location. The VSE estimation therefore can be carried out with the available measurement data only without requiring the information on operating condition at that time. In doing so the concept of system area mapping is used for the first time in combination with CNNs to explore the way of presenting system configuration in data structure and use it for sag estimation at system level. Finally, the paper establishes that the selection of variables included in the input features significantly affects the VSE performance hence a practical guidance is provided on the selection of input feature combination for the problem to be solved. The approach is applied and illustrated on the IEEE 68-bus test network.

II. Voltage Sag Estimation Using Deep Learning

A. Problem formulation

The voltage sag estimation (VSE) problem discussed here aims at estimating and classifying the sag magnitudes at unmonitored buses into six possible categories with specified voltage ranges given in Table I. Furthermore, one important feature of this problem is that the estimation is carried out considering various uncertainties associated with power system operation, e.g. renewable power injection, varying loading and load consumption, fault characteristics, etc. VSE problem is therefore to estimate the voltage ranges of sags at the unmonitored buses using phase voltages (pre-fault and during-fault voltages) at a limited number of monitored buses, as illustrated in Fig. 1. The VSE problem here is to associate the voltages at the monitored buses with the sag categories at unmonitored buses. In this way the sag categories at the unmonitored buses can be estimated directly from voltage measurement. This is essentially a classification problem, or more specifically, a supervised learning classification task to present a ground truth, i.e., to establish a function that associates the given inputs with desired outputs.

TABLE I
Definition of voltage ranges for six sag categories

Category	V-1	V-2	V-3	V-4	V-5	V-6
Ranges (p.u.)	≥0. 9	≥0.8, <0.9	≥0.7, <0.8	≥0.6, <0.7	≥0.5, <0.6	<0.5

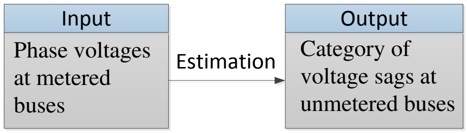

Input
Phase voltages at metered buses
Estimation
Output
Category of voltage sags at unmetered buses

Fig. 1. Illustration of the input and output prediction for VSE problem.

B. Methodology

In this study, a convolutional neural network (CNN) based approach is developed to perform classification. Data is separated into training (85% or X samples) and validation data (15% or Y samples). The network is repeatedly trained with training data samples so that it can continuously adjust its internal weights to match the input with its respective category. After training, the network is then used to classify or to predict the category for the unseen validation data. The obtained classification results will then be compared with the expected outputs to validate the CNN network performance.

The selection of input features to CNNs, as well as the construction/format of the input data, will impact estimation performance, thus they should be properly selected/designed. For the system level estimation problem discussed here, the raw data (i.e., phase voltages) should be re-structured in a way that the power network configuration is reflected in the input features before feeding into CNN networks, which is to take advantage of the structure recognition capability in CNNs.

1) Fault Simulation under Varying Operating Conditions

The training and validation data should be prepared before applying them in learning process. The raw data is obtained from simulation carried out in commercially available software DIgSILENT/PowerFactory. A set of *N* different operating conditions are selected based on historical records of intermittent renewable injection and load demand variation. Since voltage sags are mainly caused by faults in the network [15], the faults including single line to ground fault (SLGF), line to line to ground fault (LLGF), line to line fault (LLF) and three phase fault (LLLF) are simulated at each transmission line under different operating condition, separately. The voltage profiles obtained in the simulations are then used for generating training and validation data.

2) Bus Matrix Reflecting System Configuration and Data Preparation

For each data sample (corresponding to one simulation under one specified operation condition), the phase voltages are used to construct the input features, i.e., inputs to CNN. The input features consist of the following three variables:

- The voltage reduction (voltage drop) for the most severe sag at metered buses. The voltage drop, denoted as V_{red}, is calculated as $V_{pre\text{-}fault}$ -V_{sag}, where $V_{pre\text{-}fault}$ and V_{sag} are the pre-fault and during-fault voltages of the most severe sag at a bus respectively. Pre-fault voltages can either be obtained by direct measurement if a monitor for this purpose is present, or by estimating the voltage using power system state estimation [22]. V_{red} reflects sag severity and it is useful for the learning process to build up the hidden links between sags and sag profiles.

- The most severe voltage sag at the metered buses, i.e., V_{sag}. The purpose of taking V_{sag} as one of the input features is to address the VSE objective which is to estimate V_{sag} (voltage sag magnitude) at the unmonitored bus.
- The during-fault negative sequence voltage V_2.

The aforementioned three variables measured at the metered buses are used to construct three matrices respectively, which will be used as input matrices for learning process. The selection of input variables, as one of the critical steps that determines the performance of the proposed approach, will be further explained in Section III-D.

Before constructing the input matrices for CNN networks, bus matrix is formed according to the topological locations of the metered buses in order to reflect the system configuration. The bus matrix can be formed by the following procedure:

1. The network is divided into smaller areas according to its topology. For some networks, areas are already clearly defined geographically with the interconnection of inter-area ac ties, e.g., the test network used in Section III. With the area division, the meters located in the same area are clustered as one group of meters.
2. The meters are allocated in patches in the bus matrix following two rules: the meters clustered as one group should be closely allocated in a patch in the bus matrix; and the meter groups which are from the neighboring areas should be allocated next to each other in the bus matrix.
3. Steps 1-2 are repeated until the locations of all meters are determined in the bus matrix. If the number of meters is smaller than that of cells in the bus matrix, the empty cells can be filled with zero.

Taking the 24-bus network as example, assume there are 9 meters distributed around the network, as shown in Fig. 2(a), where the metered buses are highlighted in red. According to step 1, the network is topologically divided into smaller areas. 3 smaller areas are obtained, as separated by the blue dashed lines in Fig. 2(a). Based on the area division, the meters are clustered into 3 groups, (2, 3, 15, 24), (6, 10) and (1, 5, 13). Based on the two rules introduced in step 2, the meter groups are allocated in patches in the bus matrix as shown in Fig. 2(a).

There are in total four meters in group 1. If the arrangement of the four meters in the corresponding patch is still not clear, step 3 can be performed to further determine the arrangement of meters in group 1. In this way, the area corresponding to group 1 is divided into smaller sub-areas, as shown in Fig. 2(b). It can be seen that by dividing the area into sub-areas, the arrangement of the meters in the bus matrix can be easily identified. With the same procedures applied to the other two areas, the location of all meters in the bus matrix can be determined as given in Table II. It can be seen from Table II that the bus matrix is arranged in a way that buses located topologically closer are adjacent to each other in the bus matrix.

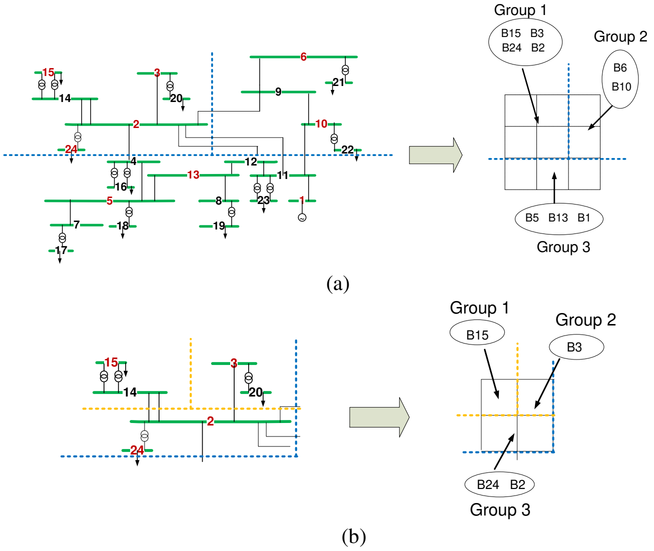


Fig. 2. Illustration of the procedure of forming bus matrix: (a) divide the network into smaller areas, (b) divide the area into smaller sub-areas

TABLE II
Illustration of the bus matrix

B15	B3	B6
B24	B2	B10
B5	B13	B1

Based on the bus matrix, three variable matrices, each corresponding to one of the aforementioned three variables, are constructed for each data sample, as illustrated in the input matrix in Fig. 3. It can be seen that the input to CNN network is a rectangular volume whose width and height (i.e., the vertical and horizontal dimensions of the input matrices) are determined by the bus matrix, while the depth is defined as three layers, with one stacked on top of the other.

3) Correlation and Convolution of Variables among Buses

Unlike regular neural networks which connect all input features together at a time, the input features here go through a number of convolution layers first before being connected to a fully-connected neural network. On the other hand, instead of focusing on one variable at one bus, the convolutional layer takes into account three variables at multiple physically close buses at one time via kernels/filters. The kernel is a rectangular volume with width and height smaller than that in input matrix and its depth equal to that in input matrix, as shown in Fig. 3. Each dimension of the kernel is usually a square patch. The convolution layer takes in a rectangular volume (equal in size to the kernel) from the input matrix and.

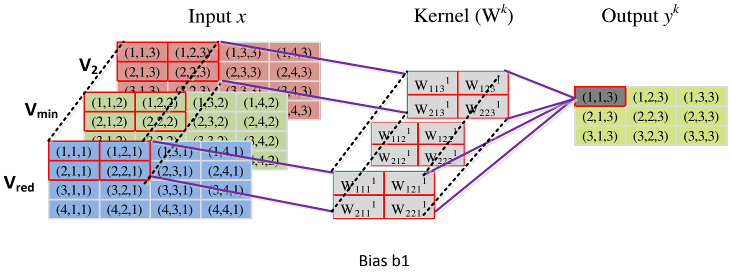


Fig. 3. Illustration of the process of one convolution layer with one kernel.

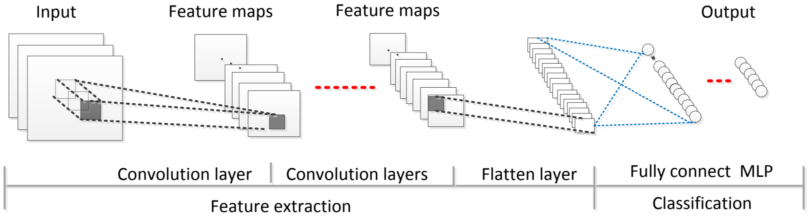


Fig. 4. Illustration of the process of deep learning used for VSE.

passes them through the kernel using dot product. As illustrated in Fig. 3, with the *k*th kernel W^k, each node in the activation map y^k (or named as activation map) is calculated according to:

$$y_{i,j}^{k} = \sigma\left(\sum_{d=1}^{D}\sum_{r=1}^{F}\sum_{c=1}^{F} x_{(r+i\times S),(c+j\times S),d} \times w_{r,c,d}^{k} + b^{k}\right) \quad (1)$$

$$0 \le i \le \frac{H-F}{S};\ 0 \le j \le \frac{W-F}{S}$$

where $y_{i,j}^{k}$ denotes the output value of a node on the feature map for kernel *k*; *H*, *W* and *D* stand for height, width and depth of the input volume; *F* denotes the height and width size of the kernel and *S* stands for the stride length, i.e, the step size of the filter's movement across the input volume; $w_{r,c,d}^{k}$ denotes the weight positioned at (*r*,*c*,*d*) in kernel *k*; b^k is the bias at kernel *k*. The term σ denotes activation function which is to non-linearize the linear convolution operation. Rectified linear unit (ReLU) is used as the activation function for convolution layers. The aggregated value in a way reflects the correlation/relation among three variables at four buses covered (as shown in Fig. 3). It represents the pattern feature that is extracted from the small rectangular volume of the input matrix against the pattern stored in the kernel.

The kernel will slide through the input volume horizontally and vertically, in search of the patterns among different variables and among different buses [23]. In a feature map, all nodes are obtained based on the same kernel, attempting to find a feature of similar characteristic, i.e., the inter-variable correlation locally among different buses. Usually a number of kernels are used for one convolution layer to extract different features from the same input.

4) Process of Deep Learning

The process of deep learning used for VSE is illustrated in Fig. 4. With a number of kernels, the convolution layers will generate a stack of activation/feature maps, as shown in Fig. 4. With a series of convolution layers, the features are extracted locally and gradually globally as search proceeds in different convolution layers. Following the convolution layers, a flatten layer is used to convert the feature maps into one vector so the fully-connected multiple layer neural network can be applied afterwards.

Since deep neural networks have a large number of parameters, such large networks usually may encounter overfitting issues. Furthermore large networks also reduce the learning and prediction speed. Dropout is used to address this issue. It randomly drops units which have weights below the threshold (along with their connections) from the neural network during training. This prevents units from co-adapting too much [24]. For each layer of the fully-connected layer, dropout is applied to avoid over-fitting.

When the training set is very large, evaluating the sums of gradients becomes very expensive as evaluating the gradient requires evaluating all the summand functions' gradients. To resolve that, stochastic gradient descent approach is adopted. It samples a subset of summand functions at a time and update weights once. The number of samples in the subset is defined as batch size. This approach is usually very effective in the case of large-scale machine learning problems.

5) System Area Mapping and Feature Extraction

The process of feature extraction from input matrices is analysed from the perspective of power system configuration. Assume convolution is applied to extract features from the input data which are obtained from a 24-bus test network as shown in Fig. 5. As mentioned in Section II-B-2, the reflection of power system configuration in the input matrices is implemented through bus matrix. The bus matrix is formed by 9 buses and the kernel is a square patch of 2×2 (for simplicity of understanding, the depth of the kernel is set to 1 and only one kernel is used in the following illustration). Firstly, the kernel takes in the first square patch from the input matrix (corresponding to Buses 15, 3, 24 and 2 as shown in the bus matrix in Fig. 5). This square patch maps the area of Loc1 in the power system, as shown in Fig. 5. With the dot product applied to this square patch of the input matrix and the kernel, the feature/characteristic in local area Loc1 is extracted and integrated as A1 in the feature map, as shown in Fig. 5. When the kernel slides through the input matrix, the features extracted from different local areas (such as local areas Loc2, Loc3 and Loc4) are obtained and used to construct the feature map. The stride size of the filter's movement across the input is set to one in the study due to the small size of the input matrices (i.e., small number of metered buses). Besides, it also avoids the case that some areas are not covered by the processing of feature extraction.

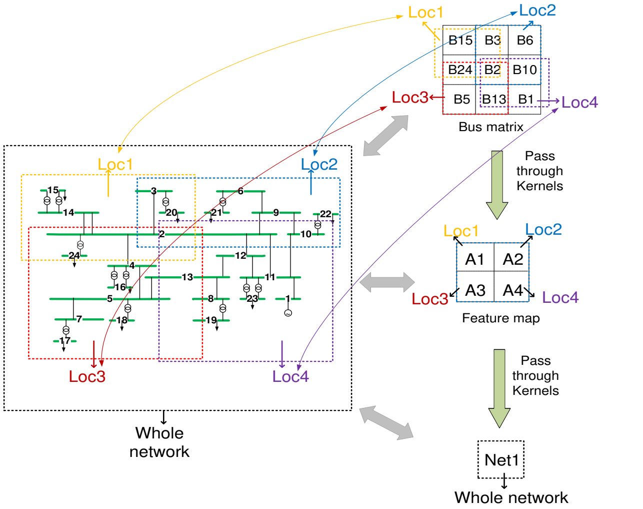


Fig. 5. Illustration of the process of feature extraction from 24-bus test network.

After obtaining the feature representation for various local areas (i.e., the feature map in Fig. 5), the feature map will pass through another kernel in the next convolution layer, which integrates the feature representations obtained from Loc1, Loc2, Loc3 and Loc4 respectively. In this way, the global integrated value, i.e., Net 1 in Fig. 5, to some extent represents the feature extracted from the whole network. One kernel is used for the illustration here. However, in general a large number of kernels are used to extract features from different perspectives in order to address the variety of characteristics and patterns existing in the network as discussed above.

6) *Update of Learning Model*

The kernel pattern in convolutional layers is stored in the format of weights, and the weights are updated gradually during learning/training in order to achieve the optimal patterns that can best distinguish the input features of different sag categories. Apart from that, the weights in fully-connected neural networks are also updated during training process in order to achieve the optimal weights which provide the best classification performance. The aforementioned weights are updated using optimiser RMSprop [25], which is a gradient descent optimisation algorithm with adaptive learning rate. The weight/parameter θ is updated according to:

$$E[g^2]_t = 0.9E[g^2]_{t-1} + 0.1E[g^2]_t \qquad (2)$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} g_t \qquad (3)$$

where $E[g^2]_t$ denotes the average of the squared gradient g_t at time step t; η is the learning rate (set to 0.0001 in the study); ϵ is a smoothing term used to avoid division by zero (set to 1e−6). The gradient estimation of (2) can be seen as gradient estimation with momentum, with 0.9 weight applied to the observed gradient of previous interaction and 0.1 weight to the gradient observed at current iteration.

III. Results of Simulation And Analysis

A. *Test System Modeling and Deep Learning Model*

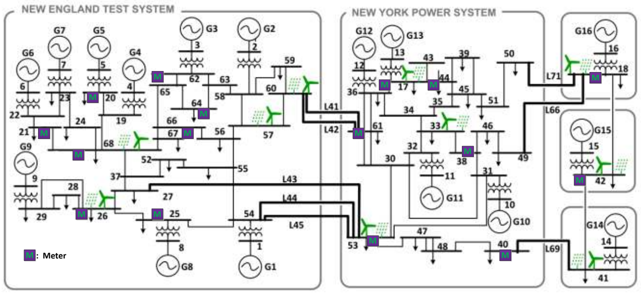


Fig. 6. Modified IEEE 68 bus test network.

The modified IEEE 68-bus test network as given in Fig. 6 presents a realistic complex meshed transmission network and has been used for various power systems studies in the past [26]. The network has five distinct areas interconnected with inter-area tie lines, which is suitable for testing the concept of system area mapping proposed in the paper. Apart from the 10 machines in the original network, 20 distributed generators (DGs) consisting of 10 wind generators modeled with DFIGs and 10 PVs, are integrated in the network. 16 meters (dark coloured squares in Fig. 6) are randomly distributed around the network as shown in Fig. 6. The bus matrix of metered buses which is constructed based on the rule introduced in Section II-B is provided in Table III. The sag performance at a non-metered bus B54 is estimated in the study.

TABLE III
Bus matrix constructed based on Fig. 6

B20	B62	B17	B18
B68	B64	B44	B42
B21	B66	B61	B38
B26	B25	B53	B40

In total 1300 operating conditions are considered and numerous faults in the system are simulated for each of them to generate voltage sags at system buses. The variation of the hourly load demands of different types (including commercial, industrial and residential loads) is adopted from the data extracted from the 2010 survey [27]. The hourly outputs of wind and photovoltaic generators are based on realistic output data considering the UK weather [28]. Four types of faults, including SLGF, LLGF, LLF and LLLF, are simulated for each transmission line under each operating condition individually. There are 72 transmission lines in the network. Thus in total 1,300×4×72= 374,400 faults are simulated to generate voltage sag the training and validation data. The phase voltages obtained in one simulation are used to generate one data sample according to Section II-B-2 (one data sample consists of the input features and their expected output which are obtained by one simulation). Although more than 250,000 samples are obtained for category V-1 presented in Table I, only 4,800 samples are selected for training to avoid over-representation. The samples are split into two groups and used for training and validation respectively, as presented in Table IV. For each sag category, about 15% of the samples are used for the validation/prediction (without being used in training process). It can be seen from Table IV that smaller size of samples is obtained for the sag categories with lower magnitudes, and that for some categories there are no samples at all (Bus B54 is a strong bus with a generator connected to it so it is capable of maintaining voltage to reasonably high values for external faults).

TABLE IV

Data size for training and validation

Type of Fault		Voltage sag magnitude					
		≥0. 9	≥0.8, <0.9	≥0.7, <0.8	≥0.6, <0.7	≥0.5, <0.6	<0.5
SLGF	Training	1000	1000	800	800	0	0
	validation	200	200	200	200	0	0
LLLF	Training	1000	1000	1000	0	800	800
	validation	200	200	200	0	200	200
LLF	Training	1000	1000	1000	1000	0	0
	validation	200	200	200	200	0	0
LLGF	Training	1000	1000	0	1000	1000	0
	validation	200	200	0	200	200	0

In the study, two performance metrics are used to evaluate the VSE estimation performance. The accuracy, as defined in (4), provides the percentage of the correct predictions over the total predictions made. It has been widely used to measure the estimation preformation in literature.

$$\text{Accuracy (\%)} = \frac{correct\ predictions}{total\ predictions} \times 100 \tag{4}$$

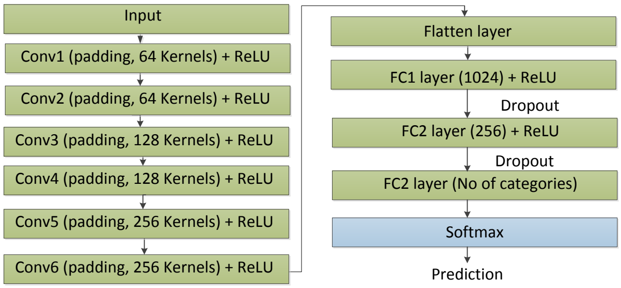
Categorical Cross-Entropy Loss, also interpreted as a cost function [18], is defined as below:

$$L_{cross-entropy} = -\sum_{n=1}^{N_S}\sum_{i=1}^{N_C} y_{ni}\log(\hat{y}_{ni}) \tag{5}$$

where N_S and N_C are the number of samples and categories respectively. y_{ni} represents whether sample n is classified in the expected/correct category i: y_{ni}=1 means that the sample n should belong to this category i, otherwise y_{ni}=0. With the use of softmax activation function for the output neurons in the model, $\hat{y}_{ni}$ can be interpreted as the probability of sample n being classified in category i. $L_{cross-entropy}$ tends toward zero as the estimated value becomes more certain to be classified in the expected class. Since validation data is not included in training process, the accuracy and loss obtained from the validation data are used to present the VSE performance as it shows the estimation performance when classifying unaccounted input samples.

The deep learning model developed for VSE in the study is the finely tuned derivative of VGG16 architecture which has been widely adopted in the past for various applications [29], as given in Fig. 7. Six convolution layers with 64, 64, 128, 128, 256 and 256 kernels respectively are used in the model. Usually max pooling is an important component in convolutional neural networks in order to down-sample an input representation for image classification. However, for the developed learning model, the size of the input representation and hidden-layer feature maps are relatively small due to the small number of monitors placed in the network, thus down-sample discretization process is not adopted in this case. Padding in Fig. 7 is to pad the input volume with zeros around the border to enable the control of the spatial size of the output volumes of convolutional layers. Dropout of 0.01 is applied in FC neural networks. The batch size is set to 10.

With the optimiser introduced in Section II-B-5, the model is trained with a large amount of training data by optimising the weights in kernels and FC neural networks until satisfactory estimation performance is achieved.

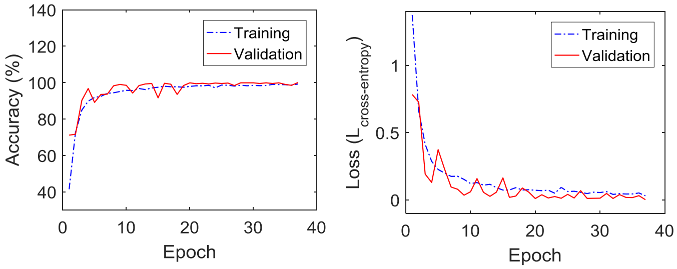


Input
Conv1 (padding, 64 Kernels) + ReLU
Conv2 (padding, 64 Kernels) + ReLU
Conv3 (padding, 128 Kernels) + ReLU
Conv4 (padding, 128 Kernels) + ReLU
Conv5 (padding, 256 Kernels) + ReLU
Conv6 (padding, 256 Kernels) + ReLU
Flatten layer
FC1 layer (1024) + ReLU
Dropout
FC2 layer (256) + ReLU
Dropout
FC2 layer (No of categories)
Softmax
Prediction

Fig. 7. Deep learning model proposed for VSE.

B. Simulation Results

The convergence of accuracy and $L_{cross-entropy}$ are presented in Fig. 8. It can be seen from Fig. 8 (a) that both training and validation have similar convergence characteristics with steady increasing accuracy along the training process. The accuracy of validation reaches 100% with 37 epochs/iterations. Fig. 8 (b) shows that both losses obtained from training and validation decrease gradually and that the training loss is relatively larger. This is expected as $L_{cross-entropy}$ is the integrated loss among all samples, and much more samples are employed in training than in validation. The validation $L_{cross-entropy}$ reaches the value of 0.001 after 37 epochs, so there is a matching, of acceptable accuracy, between the estimated outputs and the expected/correct outputs (Note: If the training continues up to 60 epochs, the validation $L_{cross-entropy}$ = 4.38×10^{-5}). To ensure consistency and robustness of results the training is repeated 50 times. The mean and standard deviation of the accuracy for the training data are 99.47% and 0.13%, respectively, hence, good performance is consistently achieved.

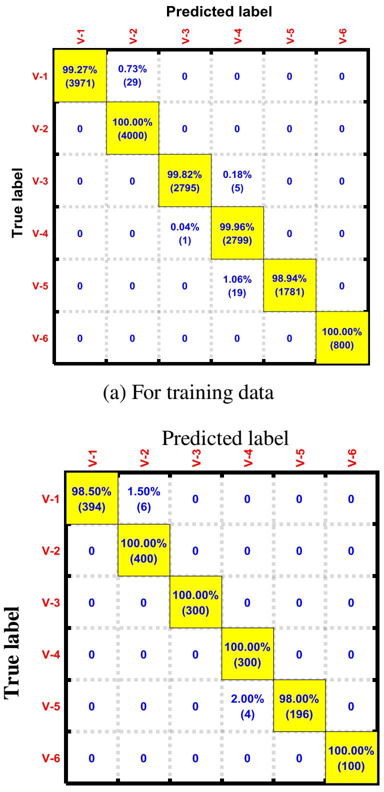


(a) Accuracy (b) $L_{cross-entropy}$

Fig. 8. Convergence of VSE performance.

The confusion matrix [30] is widely used for presenting more detailed information if the data set is unbalanced, i.e., the number of samples in different classes vary. The confusion matrix of the training data obtained with the best trained model is given in Fig. 9(a), with overall accuracy of 99.67%. It shows that a small percentage of inaccurate prediction occurs at V-1, V-3, V-4 and V-5, and the predicted category is the one next to the expected, due to the overlap of parameters between two neighboring classes and existence of samples with values very close to the class boundary.

In order to further test the obtained model, another set of 1700 samples, which has not been used for training nor for validation, is used for cross validation. An overall estimation accuracy of 99.41% is obtained in this case. The confusion matrix obtained from this set of data is given in Fig. 9(b) and shows that the four out of six sag categories can be predicted with 100% accuracy while the accuracy of the other two is above 98%. The fifth row in Fig. 9(b) suggests that among the 1700 unseen samples, there should be in total 4+196=200 sags located within the voltage range of [0.5, 0.6). However 196 out of 200 are correctly identified, and the estimated voltage range for the other four is [0.4, 0.5), which is 10% deviation from the expected range. The fourth column in Fig. 9(b) suggests that for the predicted sag category v-4, i.e., voltage range of [0.4, 0.5), the estimated voltage ranges are correctly identified with 98.68% (300/(300+4)%) certainty, while in 1.32% of the cases the predicted voltage range should be [0.5, 0.6), which is 10% deviation from the estimated range [0.4, 0.5). It can be seen therefore that the confusion matrix also reveals the distribution of the faulted estimation of sag categories.



(a) For training data

(b) For test data

Fig. 9. Confusion matrix.

The measurement uncertainty of RMS value of voltage is tested by including noise in the inputs of unseen data (i.e., the data are not used for training) based on IEC61000-4-30: class A performance [31]. The overall accuracy of 98.13% of VSE is obtained in this case, i.e., still very high. Although the performance is slightly worse than the results obtained with the ideal data shown in Fig. 8, all the faulty estimation is located at the neighbouring classes.

To visually present the kernels/filters trained during the learning process, the 64 kernels at the first convolution layer used to extract the features from local areas in the network (i.e., those used to process the inputs: V_{red}, $V_{\min}$ and V_2) are presented using color-map in Fig. 10. The kernels are a rectangular 3-D "objects", however they are plotted here as a 2-D color-map for the purpose of visualization. It can be seen that the 64 kernels are different from each other and present different patterns. Even within a kernel, the weights used to deal with different variables are different. This addresses the different pattern calculation among different variables.

With the trained model, once it is available, sag frequency can be easily predicted based on recorded phase voltages. Features extracted from the recorded phase voltages can be directly fed to the trained model to generate estimation results without considering the operating conditions under which the recorded sags occurred. With the advantage of quick estimation, this approach has the potential to be used for the cases when estimation time is constrained. As long as the monitored voltage data are achieved, the estimation results can be generated in no time. The approach provides sag categories of voltage ranges in addition to sag frequency if recorded data are available, while no need for detailed modelling information during estimation.

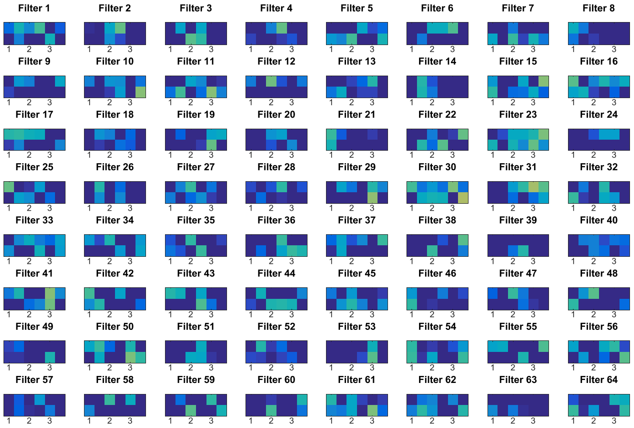


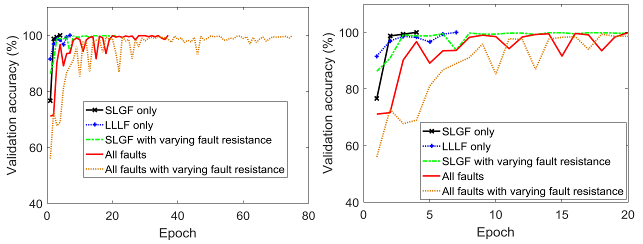
Fig. 10. Kernels of the first convolution layer obtained with 37 epochs.

C. *Comparison of Estimation Performance in Solving Different Scales of Problems*

The performance of the deep learning model in solving different scale/complexity of VSE problems is analysed further using the following five cases:

- Case 1: SLGF considering varying operating conditions discussed in Section II-B. 1300 different operating conditions are used and 1,300×72 faults are simulated to generate the input data for training and validation (see Section III-A). 3600 samples are selected for training and 800 for validation (the two rows corresponding to SLGF in Table IV).
- Case 2: LLLF considering varying operating conditions (the two rows corresponding to LLLF in Table IV).
- Case 3: SLGF considering varying operating conditions and uncertain fault resistances R_F (0Ω, 2.5Ω and 5Ω). 1,300×72×3= 280,800 faults are simulated to generate training and validation data.
- Case 4: Four different types of faults as implemented in section III-B.
- Case 5: Same as case 4 with addition of accounting for various fault resistances. This is the case with the largest number of training data.

Cases 1-3 focus on SLGF and LLLF as most faults on transmission lines (70%-80%) are unsymmetrical SLGF [32]. The LLLF is the most severe fault though only represents about 5% of the total faults. The performance of the deep learning model obtained from the five cases is presented in Fig. 11. In the simulation the training process terminates if the validation accuracy reaches 100%. Cases 1 and 2 require 4 and 7 epochs respectively to achieve 100% estimation accuracy. It can be seen that if only one type of faults is considered, 100% accuracy of classification can be achieved with only a few epochs. It suggests the relative simplicity of the classification problems in cases 1 and 2 even though the operating condition (load variation and intermittent renewable energy output) is uncertain. Cases 3 and 4 require 21 and 37 epochs respectively to achieve 100% validation accuracy. As for the most comprehensive case 5, 75 epochs are needed.



(a) Epoch 1-80 (b) Epoch 1-20

Fig. 11. Convergence of accuracy obtained from the five cases with different scales of VSE problems.

Generally fault allocation based sag estimation approaches developed in the literature require a series of sub-processes, including fault classification, fault location and then sag estimation. For instance in [33] the fault areas are identified first, followed by the identification of fault lines. Different from fault allocation based approaches, the VSE approach developed here can generate the results by only feeding the network with required measurement data once the trained network is available. The approach does not need to go through different derivation steps in order to achieve the final solution. Using one integrated deep learning process, the proposed approach can capture the relationship between the inputs (voltage measurement) and outputs (sag categories) under uncertain operating conditions. The sag categories can be accurately estimated regardless of the uncertainty of load demand, DG outputs, fault types, fault area and fault location. Especially for case 5, classification can be accurately performed without knowing the condition of uncertain factors including the variation of load consumption, DG outputs and fault resistance.

D. Comparison among Different Input Feature Combinations

The selection of input features no doubt impacts the sag estimation performance significantly. To analyse the impact, the set of SLGF data is used to test different combinations of input variables. Five cases of different input feature combination are selected here, as defined in Table V, where V_1 and V_0 denote during-fault positive-sequence and zero-sequence voltage respectively; and V_a, V_b and V_c are voltages at phase A, B and C respectively. The variables given in Case 1 in Table V is the same as the input features introduced in Section II. Cases 1-3 include the sag details obtained at the metered buses, while cases 4 and 5 adopt sequence voltage and phase voltage respectively. In cases 1-3, pre-fault voltages are not included in the input matrices.

TABLE V
Five cases of different input features

Cases	input features
1	$V_{pre\text{-}fault}$ -V_{sag}; V_{sag}; V_2
2	$V_{pre\text{-}fault}$ -$V_{1,}$; V_{sag}; $V_{2.}$
3	V_1; V_2; $V_{sag.}$
4	V_1; V_2; V_0
5	V_a; V_b; V_c

The performance of the estimation accuracy obtained in the five cases is presented in Fig. 12. In the simulation the training process terminates when the validation accuracy reaches 100% except for case 5 which can only reach maximum of 17.7% accuracy. It can be seen from case 5 that if phase voltages are used directly as the input features the sag categories cannot be distinguished. Among cases 1-4, case 1 presents the best convergence characteristic and only 4 epochs are required, while cases 3, 4 and 2 require 7, 11 and 22 epochs respectively in order to achieve 100% accuracy. In case 3, even though the pre-fault voltage is not included, its performance is ranked as the second among the five cases. As for case 4, although the sag information and pre-fault voltage are not included in input features, the sag categories are still distinguishable by taking relatively more epochs. Between cases 4 and 5, it can be seen that although the input data in case 4 are derived from phase voltages (i.e., the input data in case 5), case 4 has much better performance than case 5. It suggests that different presentation of the same data source can produce completely different accuracy performance. This highlights the importance of using the right presentation (extracting the right features) of the raw data before using them for deep learning training.

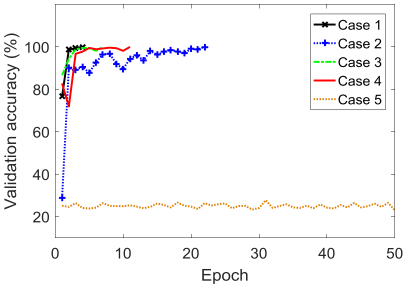


Fig. 12. Convergence of accuracy obtained from the five cases with different input features.

TABLE VI
Range of coefficient settings

			$L_{cross-entropy}$	
No.	Coeff.	Range	Mean	Std.
1	Dropout rate	[0 0.2]	0.0023	0.0022
2	Learning rate	[0.00005 0.0005]	0.0018	0.0012
3	Batch size	[1 100]	0.0021	0.0021
4	No. of nodes in FC1	[128 2048]	0.0024	0.0030

E. Comparison among Different Meter Placements and Bus Matrices

To further present the performance of VSE under different scenarios, different bus matrices and meter placements are tested for different purposes:

- MP1: Meter placement is given in Fig. 6. The observability analysis of the system with this set of meters (in the state estimation sense) is carried out using the topology-based method [34, 35]. The unobservable buses were B1, B4, B6, B7, B9-B12, B14, B23, B24, B30, B32, B34, B35, B37, B47, B52, B54, B55, B58, B59 and B63. The bus of interest, i.e., non-monitored bus B54, is unobservable. The bus matrix is constructed based on the procedure introduced in Section II-B and provided in Table III.

- MP2: The meters are the same as MP1. However the bus matrix is randomly constructed without using the proposed procedure in Section II-B. It can be seen from Fig. 6 that the two areas where B18 and B42 locate are topologically closer to each other. According to the rules introduced in Section II-B, the meters in these two areas should be adjacent to each other in the bus matrix. However, in the bus matrix for MP2 as given in Table VII, the two buses are not closer to each other. This case study is to test the impact of bus matrix construction on VSE performance.

TABLE VII

Bus matrix for MP2

B18	B20	B38	B25
B61	B62	B42	B64
B40	B53	B68	B21
B26	B17	B66	B44

- MP3: Different from MP1, the meters are selected based on transformer locations, as an example of generalized meter placement. The unobservable buses include B27, B37, B47, B48, B52, B53 and B55. The bus of interest (non-monitored bus B55) is unobservable. The bus matrix as given in Table VIII is constructed based on the proposed procedure. The cells of different color represent different areas/subareas divided in the network when applying the procedure of bus matrix construction.

TABLE VIII

Bus matrix for MP3

B19	B62	B58	B18
B20	B23	B17	B42
B29	B22	B36	B41
B25	B54	B32	B31

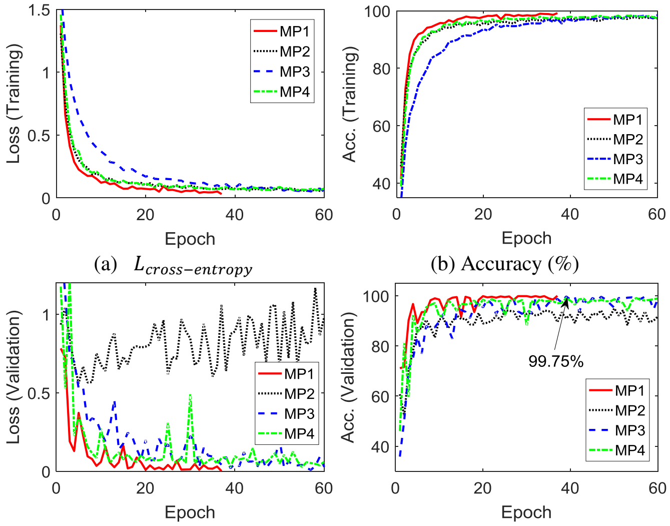
- MP4: 9 meters are selected from the 16 meters used in MP1. The unobservable buses include B1, B4, B6-B15, B18, B23, B24, B30, B32-B38, B42, B47, B49-52, B54, B55, B58-B60, B63-B65, B67 and B68. The bus of interest, i.e., non-monitored bus B54, is unobservable. The bus matrix is constructed based on the proposed procedure, as given in Table IX.

TABLE IX

Bus matrix for MP4

B20	B62	B44
B21	B66	B61
B26	B53	B40

The results are given in Fig. 13, and the detailed statistics are given in Table X. For MP2, although the accuracy and $L_{cross-entropy}$ obtained in training are acceptable compared to other scenarios, the loss and overall accuracy obtained in validation is obviously worse than other cases. It can be seen that the unseen data cannot be estimated well without using the proposed procedure to construct the bus matrix, which suggests the importance of input matrix structure. As for the generalized meter placement MP3, although its results are not as accurate as those obtained by MP1, it still generates acceptable results with maximum 99.75% accuracy for the unseen data. The use of fewer meters (i.e., MP4) still yields 99.12% accuracy for the unseen data. This is reasonable as less information is used for input and the overall performance is slightly compromised in this case. It can be clearly seen from the study that the performance can be notably improved with proper meter placement. The optimal meter placement, however, is not within the scope of this study and hence it is not discussed in detail in the paper.



(a) $L_{cross-entropy}$ (b) Accuracy (%)

(c) $L_{cross-entropy}$ (d) Accuracy (%)

Fig. 13. Convergence of accuracy obtained from different MPs.

TABLE X

Results obtained from different MPs

	Min loss (training)	Max acc. % (training)	Min loss (validation)	Max acc. % (validation)
MP1	0.0207	99.67	4.38×10^{-5}	100
MP2	0.0555	97.90	0.5597	95.00
MP3	0.0485	98.50	0.0095	99.75
MP4	0.0595	98.35	0.0272	99.12

F. Coefficient Settings in Deep Learning Model

Four coefficients in learning model, e.g., dropout rate, learning rate, batch size and number of nodes in FC1, are analysed in terms of their influence on the estimation accuracy. The ranges for these coefficients are selected based on literature [36-38] and should meet the constraint that the $L_{cross-entropy} < 0.01$. The ranges selected for the coefficients are given in Table VI. It can be seen that the learning rate has a range that is relatively smaller than that of other coefficients in order to keep loss less than 0.01. The influence of the coefficient setting is assessed in the following way. Each coefficient is set to 10 different values whilst other parameters are set as base values (The base values of the four coefficients in Table VI are set to 0.01, 0.0001, 10 and 1024 respectively). The learning models with the different coefficient settings are used to solve the problem introduced in Section III-A, and a set of 10 loss values is obtained for each coefficient. The mean and standard deviation of the loss obtained for each coefficient with 60 epochs is given in Table VI. Due to the small range of learning rate, the variation of the loss performance obtained for learning rate is also relatively smaller. The other three coefficients generate similar but still small mean and standard deviation values. The coefficient ranges given in Table VI can be used as reference for setting coefficients while solving VSE problems. In this particular study, very good performance ($L_{cross-entropy}$ =4.38×10^{-5}) was achieved with coefficients set at the base value.

IV. Conclusions

This paper proposes a CNN based voltage sag estimation approach to estimate the ranges of voltage sag magnitudes under uncertain operating condition. In the approach, the new concept of system area mapping together with system bus matrix is proposed to construct training and validation data to enable various system areas mapped to different patches in input matrices. In this way the power system configuration is embedded in the inputs for pattern learning. By sliding a number of kernels across the input matrix, convolutional layers extract the features from various local areas in power systems and then integrate them into higher levels of convolutional layers. The patterns used to classify the sag categories are stored in kernels and updated during training via optimiser RMSprop.

The simulation results demonstrated that different sag magnitude ranges can be identified accurately regardless of the operating condition during sags. The capability of the proposed approach to solve different scale of classification problems has been tested in the study. The paper also investigates the impact of input features on VSE performance, and the best combination of the input variables consists of voltage drop, the lowest voltage sag magnitude and a negative sequence voltage at metered buses. The benefit of constructing the input features based on the proposed bus matrix is analysed and it shows that the sag categories of the unseen data cannot be estimated accurately if the metered voltages are randomly located in the input matrices.

References

[1] J. Y. Chan, J. V. Milanović, and A. Delahunty, "Risk-Based Assessment of Financial Losses Due to Voltage Sag," *IEEE Trans. Power Delivery,* vol. 26, pp. 492-500, 2011.

[2] N. C. Woolley, J. M. Avendaño-Mora, and J. V. Milanović, "Methodology for robust monitoring of voltage sags based on equipment trip probabilities," *Ele. Power Syst. Res.,* vol. 90, pp. 107-116, 2012.

[3] G. Olguin, "Voltage Dip (Sag) Estimation in Power Systems based on Stochastic Assessment and Optimal Monitoring," Ph.D., Chalmers University of Technology, Sweden, 2005.

[4] N. C. Woolley, M. Avendaño-Mora, A. P. Woolley, R. Preece, and J. V. Milanovic, "Probabilistic estimation of voltage sags using erroneous measurement information," *Elec. Power Syst. Res.,* vol. 106, pp. 142–150, 2014.

[5] M. Avendaño-Mora and J. V. Milanović, "Generalized formulation of the optimal monitor placement problem for fault location," *Ele. Power Syst. Res.,* vol. 93, pp. 120-126, 2012.

[6] M. Avendano-Mora and J. V. Milanovic, "Monitor placement for reliable estimation of voltage sags in power networks," *IEEE Trans. on Power Del.,* vol. 27, pp. 936-944, 2012.

[7] I. B. N. C. Cruz, A. P. Lavega, and J. R. C. Orillaza, "Overview of methods for voltage sag performance estimation," in *17th Int. Conf. on Har. and Quality of Power (ICHQP)*, 2016, pp. 508-512.

[8] G. Ye, Y. Xiang, M. Nijhuis, V. Cuk, and J. F. G. Cobben, "Bayesian-Inference-Based Voltage Dip State Estimation," *IEEE Trans. on Instr. and Meas.,* vol. 66, pp. 2977-2987, 2017.

[9] M. H. J. Bollen, "Fast assessment methods for voltage sags in distribution systems," *IEEE Trans. on Ind. App.,* vol. 32, pp. 1414-1423, 1996.

[10] W. Li, *Risk Assessment of Power Systems: Models, Methods, and Applications, 2nd ed*: IEEE: John Wiley & Sons, Inc., 2014.

[11] T. A. Short, A. Mansoor, W. Sunderman, and A. Sundaram, "Site variation and prediction of power quality," *IEEE Trans. on Power Del.,* vol. 18, pp. 1369-1375, 2003.

[12] X. Zambrano, A. Hernandez, M. Izzeddine, and R. M. D. castro, "Estimation of voltage sags from a limited set of monitors in power systems," in *2017 IEEE ManchesterPowerTech*, 2017, pp. 1-1.

[13] E. Espinosa-Juarez and A. HernÁndez, "A method for voltage sag state estimation in power systems," *IEEE Trans. on Power Del.,* vol. 22, pp. 2517-2526, 2007.

[14] E. Espinosa-Juarez and A. Hernandez, "Neural Networks Applied to Solve the Voltage Sag State Estimation Problem: An Approach Based on the Fault Positions Concept," in *Proc. 2009 Elec., Robo. and Auto. Mech. Conf. (CERMA)*, 2009, pp. 88-93.

[15] M. H. J. Bollen, *Understanding Power Quality Problems: Voltage Sags and Interruptions*. New York: Wiley, 2000.

[16] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. of the IEEE,* vol. 86, pp. 2278-2324, 1998.

[17] A. Krizhevsk, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1-9.

[18] M. A. Nielson, *Neural Networks and Deep Learning*: Determination press, 2015.

[19] P. P. Brahma, D. Wu, and Y. She, "Why deep learning works: a manifold disentanglement perspective," *IEEE Trans. on Neural Net. and Lear. Syst.,* vol. 27, pp. 1997-2008, 2016.

[20] E. D. Varga, S. F. Beretka, C. Noce, and G. Sapienza, "Robust real-time load profile encoding and classification framework for efficient power systems operation," *IEEE Trans. on Power Syst.,* vol. 30, pp. 1897-1904, 2015.

[21] B. J. Claessens, P. Vrancx, and F. Ruelens, "Convolutional neural networks for automatic state-time feature extraction in reinforcement learning applied to residential load Control," *IEEE Trans. on Smart Grid,* vol. PP, pp. 1-1, 2016.

[22] N. C. Woolley and J. V. Milanović, "Statistical estimation of the source and level of voltage unbalance in distribution networks," *IEEE Trans. Power Del.,* vol. 27, pp. 1450-1460, 2012.

[23] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*: MIT Press, 2016.

[24] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, " Dropout: a simple way to prevent neural network from overfitting," *J. of Mach. lear.,* vol. 15, pp. 1929-1958, 2014.

[25] G. Hinton, Neural network for machine learning, lecture 6c, the momentum method [Online]. Available: http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf.

[26] B. Pal and B. Chaudhuri, *Robust control in power Systems*. New York. NY. USA: Springer, 2005.

[27] S. Hesmondhalgh, "GB energy demand-2010 and 2025. Initial brattle electricity demand-side model-scope for demand reduction and flexible response," 2012.

[28] CM SAF JRC EUROPEAN COMMISSION. Photovoltaic Geographical Information System - Interactive Maps [Online]. Available: http://re.jrc.ec.europa.eu/pvgis/apps4/pvest.php.

[29] K.Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *CoRR,* vol. abs/1409.1556, 2014.

[30] T. Fawcett, "An introduction to ROC analysis," *Pattern Rec. Letters,* vol. 27, pp. 861-874, 2006.

[31] IEC 61000-4-30: 2003 "Testing and measurement techniques – Power quality measurement methods" 2003.

[32] J. J. Grainger and W. D. Stevenson, *Power System Analysis*. New York: McGraw-Hill, 1994.

[33] M. Majidi, M. Etezadi-Amoli, and M. S. Fadali, "A sparse-data-driven approach for fault location in transmission networks," *IEEE Trans. on Smart Grid,* vol. 8, pp. 548-556, 2017.

[34] X. B. and A. A., "Optimal placement of phasor measurement units for state estimation, Final project report," PSERC, Ithaca, NY2005.

[35] J. M. A. Mora, "Monitor Placement for Estimaton of Voltage Sags in Power Systems," Ph.D. dissertation, Sch. of Electr. and Electro. Eng., University of Manchester, 2012.

[36] G. Kang, J. Li, and D. Tao, "Shakeout: a new approach to regularized deep neural network training," *IEEE Trans. on Pattern Analysis and Mach. Int.,* vol. PP, pp. 1-1, 2017.

[37] Z. Li, B. Gong, and T. Yang, "Improved dropout for shallow and deep learning," in *Proc. Advances in Neural Inf. Proc. Syst.*, 2016, pp. 2523-2531.

[38] L. W. Kim, "DeepX: deep learning accelerator for restricted boltzmann machine artificial neural networks," *IEEE Trans. on Neural Net. and Lear. Syst.,* vol. PP, pp. 1-13, 2017.