Online Probabilistic Power Flow with Tie-line Power Transfer: A Stacked Denoising Auto-Encoders Method

Shaoqun Song¹, Jiayue Zhan², Zhixuan Liu¹, Xin Cheng¹, Xiaowen Lai³*
¹ State Grid Fujian Electric Power Co., Ltd., Fuzhou 350001, China
² Materials branch of State Grid Fujian Electric Power Co., Ltd, Fuzhou 350001, China
³ Beijing TsIntergy Technology Co., Ltd., Beijing 100080, China
*Corresponding author’s e-mail: liuchong@tsintergy.com

Abstract. Based on stacked denoising auto-encoders (SDAE) and combined with the Monte-Carlo simulation (MCS) method, this paper proposes a hybrid algorithm (SDAE-Based-PPF, SPPF) for online calculation of probabilistic power flow (PPF) with tie-line power transfer. By incorporating the deep structure and reconstructive strategy of SDAE, this paper extracts high-level features of nonlinear power flow equations, thereby constructing SDAE-based power flow models. Then, the parameters of SDAE-based power flow models, with improved training speed and accuracy, are optimized by mini-batch gradient descent algorithm, momentum learning rate and introduced cross-entropy loss function. The SDAE-based power flow models are developed to approximate power flow equations with unidirectional extracted high-level features, so as to enable non-iterative solvability judgment and calculation of power flow, therefore fast and accurate. Benefiting from the parallelizability, speed and accuracy of SDAE, the proposed SDAE-based power flow models, with unsolved random samples generated by the MCS method, are able to calculate all samples simultaneously, and this enables PPF to be calculated online. Finally, numerical results are implemented on three standard IEEE test cases to verify the effectiveness of proposed SDAE-based power flow models and SPPF method.

1. Introduction
Interconnected power systems improve utilization efficiency of power resources by allowing tie-line power transfer among different regional power systems. However, power systems are affected by uncertainties, including power demand and renewable energy sources. This leads to the uncertainties of tie-line power transfer. Under such a case, probabilistic power flow is fundamental to analyze the uncertainties and probabilistic characteristics of state variables (e.g., tie-line power transfer) in power systems, and thus has attracted wide attention in areas such as power systems planning and operation [1]-[3]. In recent years, due to the integration of intermittent renewable energy systems containing photovoltaic energy and wind energy, the uncertainties in power systems are increasingly obvious. In order to timely provide basis for power system economic dispatch [4] and security analysis [5], a practicable methodology that could compute the probabilistic power flow online becomes preferable.

The area of probabilistic power flow has spanned over forty years since proposed by Borkowska [1]. Currently, three approaches are popular in computing probabilistic power flow: analytical, approximate and simulation methods. Analytical methods, including convolution and fast Fourier transform [6], [7], are based on some assumptions and simplifications and thus benefit from less calculation burden, but...
the numerical characteristics accuracy of output variables are usually deteriorated if the number of input random variables increases [8]. Approximate methods, including the point estimate method [9] and the unscented transformation methods [10], also benefit from less calculation burden, but only statistical moments are calculated, where complete probabilistic density functions remains hard to obtain. Simulation methods are based on deterministic power flow calculation for all sampled power system status, thus considered to be more accurate compared with analytical methods and approximation methods. Among them, the Monte Carlo simulation by use of simple random sampling (MCS-SRS) is commonly used as reference to validate other methods [11], but the MCS-SRS requires a tremendous number of repeatedly solutions of deterministic power flow, and thus requires more computation time. Hence, researches have been done for reducing the computation time of the MCS method in probabilistic power flow.

Compared with traditional neural networks, regarding accuracy, deep neural networks, with multi-hidden layers structure, are able to extract high-level nonlinear features from large amount of data. Besides, deep neural networks are theoretically more efficient at representing complicated distributions [21]. As a variation of deep neural networks, stacked denoising auto-encoders (SDAE) is able to handle regression and classification problem simultaneously, and this means judgment of power flow solvability and calculation of power flow can be achieved with only a single calculation.

Consequently, this paper presents SPPF, a hybrid algorithm with combination of SDAE-based power flow models and the MCS method; the contributions of this paper are twofold:

• SDAE-based power flow models with corresponding improved training methods are proposed. Based on the approximation and solvability judgment capability of SDAE in nonlinear power flow equations, SDAE-based power flow models are constructed in this paper. After properly trained, SDAE-based power flow models are able to judge the solvability of power systems and non-iteratively calculate power flow.

• An online calculation method of probabilistic power flow based on SDAE and combined with the MCS is proposed. After random sampling with the Monte-Carlo method, the power flow of all samples is determined by SDAE-based power flow models simultaneously.

The rest of the paper is organized as follows. The SDAE-based power flow models and method to train the models are illustrated in Section 2. Probabilistic power flow method based on SDAE and combined with the MCS is presented in Section 3. Simulation results are shown in Section 4, followed by the conclusions in Section 5.

2. Power Flow Models Based on SDAE
Basic structure of SDAE is introduced in this section. Then, SDAE-based power flow models are presented, with two-stage training algorithm improved by cross-entropy loss function, mini-batch gradient descent algorithm and momentum learning rate.

2.1. Structure of stacked denoising auto-encoders
SDAE is composed with denoising auto-encoders layer by layer. As extension models of auto-encoder (AE), DAE forces AE to comprehensively extract high-level features of input $X$ by locally corrupting input vector $X$, whereby improving the robustness of models [23]. DAE is made of input layer $X$, single middle layer and output later $Z$, with those layers linked by encoder and decoder. The logic and structure diagram of DAE is illustrated in Fig.1, and its detailed computational procedure is as follows:

By corrupting initial input $X$ with random mapping in (1), locally corrupted input $\tilde{X}$ is gotten.

$$\tilde{X} \sim q_{\beta}(X)$$ (1)

where, $q_{\beta}$ denotes corruption procedure by random mapping, namely to randomly select a certain number of input variables set to zero. This procedure helps DAE adapt to errors and missing of data and improves the robustness of DAE, which will be illustrated in Section 4.

By encoding function $f_{\beta}$ in (2), middle layer $Y$ is determined by corrupted input $\tilde{X}$. 

2
\[ Y = f_\theta(\tilde{X}) = s(W\tilde{X} + b) \]  

(2)

where, weight matrix \( W \) is a \( d_x \times d_y \) matrix; biased vector \( b \) is a \( d_y \)-dimensional vector. \( d_x \) and \( d_y \) are the row dimension and column dimension of middle layer, respectively. \( s(x) = 1/(1 + e^{-x}) \) is sigmoid function, and is selected as activation function of encoding and decoding in this work.

By decoding function \( g_\theta \) in (3), output layer \( Z \) of DAE is determined by middle layer \( Y \).

\[ Z = g_\theta(Y) = s(W'Y + b') \]  

(3)

where, weight matrix of decoder \( W' \) is a \( d_y \times d_x \) matrix; biased vector \( b' \) is \( d_x \) vector.

By stacking DAE layer by layer, with the middle layer of DAE in lower layer applied as the input of DAE in higher layer, deep neural network constructed by SDAE is gotten, and its structure is illustrated in Fig. 2. Note that the output layer \( Z \) of DAE is not included in calculating the output of SDAE, and marked by grey in the figure. By extracting the high-level features of input \( X \) with continuously encoding, the output \( Y_l \) of SDAE is finally obtained, as shown in (4).

\[ Y_l = f^{(l)}_\theta(\ldots f^{(1)}_\theta(X)) \]  

(4)

where, \( f^{(l)}_\theta \) is the encoding function of \( l \)th DAE, \( l=1, 2, \ldots, n \), and \( n \) represent the number of DAE in SDAE, \( f^{(n)}_\theta \) is the encoding function of top layer.
2.2. SDAE-based power flow models and their training methods

Based on the sound performance of SDAE in approximating non-linear mapping including equations, SDAE-based power flow models are presented and shown in Fig.2. The input $X$ of SDAE-based power flow models include active and reactive node injection power of all renewable energy nodes and load nodes, active injection power and voltage magnitude of PV nodes; by equation (4), power flow results, as output $Y_t$ including solvability of power systems, node voltage of all nodes and power flow of all branches, can be determined; meanwhile, resistance, reactance of branches are not selected as input $X$, because they don’t vary with system state. Encoding function $f^{(1)}$ and $f^{(2)}$ in (4) can be seen in (2), where the optimal encoding parameters $\theta = \{W, b\}$ are the training objective of SDAE-based power flow models, including optimal weight matrix $W$ and biased vector $b$ of SDAE. Two-stage training method based on cross-entropy loss function, mini-batch gradient descent algorithm and momentum learning rate is introduced to train SDAE-based power flow models.

2.2.1. Unsupervised pre-training for SDAE-based power flow models

The optimizing objective and procedure of unsupervised pre-training SDAE is to determine optimal initial parameters $\theta = \{W, b\}$ for each DAE from bottom layer to top layer.

In the process of pre-training SDAE, cross-entropy function $L_l(x, z) = -\sum_{k=1}^{d} (x_k \log z_k + (1-x_k) \log(1-z_k))$ is introduced as loss function, and the optimizing objective function (5) is constructed by the input $X_l$ and output $Z_l$ of DAE in lth layer.

$$\arg_{\theta} \min J(W, b) = \arg_{\theta} \min L_l(X_l, Z_l)$$

(5)

where, $d$ in $L_l$ is the dimension of $x$ and $z$. $X_l$ is the input of DAE in lth layer, and also the output $Y_{l-1}$ of (l-1)th layer, obtained by corruption and encoding in (1)-(2) with initial input data $X$. $Z_l$ is the output of DAE in lth layer, obtained by corruption, encoding and decoding in (1)-(3) with $X_l$.

Then, by introducing mini-batch gradient descent algorithm into equation (5), unsupervised parameters updating equations are constructed. For each single batch, the parameters updating equations are given in (6)-(7).

$$W_{ij}^{(l, T+1)} = W_{ij}^{(l, T)} - \eta \frac{1}{m} \sum_{k=1}^{m} \frac{\partial}{\partial W_{ij}^{(l, T)}} J(W, b)$$

(6)
\[ b_i^{(l,T)} = b_i^{(l,T)} - \eta \sum_{k \in \mathcal{I}} \frac{\partial}{\partial b_i^{(l,T)}} J(W, b) \]  

(7)

Finally, in order to accelerate updating of parameters and surmount local minimum basins of attraction, momentum learning rate \( p \times dW^{(l,T-1)} \) and \( p \times db^{(l,T-1)} \) is introduced as additional item on basis of equations (6)-(7). The constructed parameters updating equations are show in (8)-(9).

\[
W^{(l,T+1)} = W^{(l,T)} + \Delta W^{(l,T)} + p \times dW^{(l,T-1)}
\]

(8)

\[
b^{(l,T+1)} = b^{(l,T)} + \Delta b^{(l,T)} + p \times db^{(l,T-1)}
\]

(9)

By the unsupervised pre-training above, the optimal parameters of encoder in each layer \( \theta = \{W, b\} \) can be determined, and as the initial parameters of supervised training in the next stage.

2.2.2. Supervised fine-tuning for SDAE-based power flow models

The optimizing objective and procedure of supervised training SDAE is to determine optimal parameters \( \theta = \{W, b\} \) for entire SDAE from the top layer to the bottom layer.

In supervised fine-tuning of SDAE, the cross-entropy loss function is constructed by the output of \( Y_t \) in SDAE \( Y_t = f^{(t)}(f^{(t-1)}(...f^{(1)}(X))) \) and the output of training sample \( Y \). The optimal target is shown in (10).

\[
\arg_\theta \min J(W, b) = \arg_\theta \min L_{\text{ce}}(Y_t, y)
\]

(10)

For equation (10), if mean square function \( L = ||x - z||^2 / 2 \) is applied as loss function, according to chain rule and mini-batch gradient descent algorithm, the determined change of weight matrix \( W^{(t)}_\theta \) in the top layer (for DAE in the top layer, this weight means jth neuron in middle layer towards ith neuron in output layer) is \( \frac{\partial J(W, b)}{\partial W^{(t)}_\theta} = (Y_{j,t} - y_j)Y_{j,t}(1-Y_{j,t}) \). Because that all elements in \( Y_{j,t} \) are between 0 and 1, \( Y_{j,t}(1-Y_{j,t}) \) is below 0.25, which leads to the decrease of efficiency in parameters updating and this is called the saturation of sigmoid function; On contrary, if cross-entropy loss function is applied, the determined change of weight matrix in top layer is \( \frac{\partial J(W, b)}{\partial W^{(t)}_\theta} = (Y_{j,t} - y_j) \), which avoids the saturation of sigmoid function and improves the efficiency of parameters updating.

On this basis, similar with the calculation procedure of unsupervised pre-training, mini-batch gradient descent algorithm (6)-(7) and momentum learning rate (8)-(9) can be applied in optimizing equation (10) and finally determine the optimal parameters of SDAE.

The flowchart of the two-stage training of SDAE power flow model above is presented in Fig. 3.

By introducing the determined optimal parameters \( \theta = \{W, b\} \) of SDAE from two-stage training into equation (2), the encoding functions \( f^{(t)}_\theta \) are determined, then all determined encoding functions are introduced into equations (4). In consequence, SDAE power flow models are constructed. The models are able to extract the high-level features of non-linear power flow equations, and replace the deterministic power flow equations with SDAE power flow models. The detailed power flow for input sample can be calculated by equation (4) efficiently; Besides, SDAE power flow models can be applied in online calculating probabilistic power flow, and the method is presented in Section 3.
3. Online probabilistic power flow based on SDAE and Combined with the MCS Method

Based on constructed SDAE power flow models (4) and corresponding training methods (5)-(10), online probabilistic power flow based on SDAE and combined with the MCS (SPPF) is proposed, and the procedure are as follow:

**Step 1)** Power flow training data acquisition: By channels such as operational monitoring, experiment and simulation, training sample for initializing SDAE is obtained.

**Step 2)** Data preprocessing: this step includes data pre-processing and determination of hyper-parameters for SDAE.

**Step 3)** Unsupervised pre-training of SDAE: Firstly, apply cross-entropy loss function into equation (5), and construct the loss function of DAE in the first layer with training sample input $X$. Then, with applied momentum learning rate and mini-batch gradient descent algorithm, the parameters updating equations (8)-(9) are constructed, whereby the optimal parameters for encoder of DAE in the first layer are determined. Afterwards, the output of the middle layer of DAE obtained from equation (2) is also the input of DAE in the second layer. Applying the same methods to construct cross-entropy loss function and so on, optimal encoding parameters $\beta = \{W, b\}$ for each DAE are determined from bottom to top.

**Step 4)** Supervised fine-tuning of SDAE power flow models: Firstly, apply cross-entropy loss function into equation (10), and construct the loss function of whole SDAE power flow models with the input and output of training samples. Then, mini-batch gradient descent algorithm and momentum learning rate are also used to construct the parameters updating equations (8)-(9), whereby all the optimal parameters of encoders $\beta = \{W, b\}$ in SDAE are determined; in consequence, SDAE power flow models are well-trained.

**Step 5)** Sampling of system state: The random variables of system including wind velocity, photovoltaic power, loads, active injection power and voltage magnitude of PV nodes are sampled.

**Step 6)** Online probabilistic power flow calculation with SPPF: input all the samples obtained from Step 5) into the well-trained SDAE power flow models in Step 4), determine the solvability and power flow for all samples simultaneously, and hence calculate power flow of all samples online.

**Step 7)** Probabilistic power flow index analysis: Calculate the mean value, standard variance and probability density of all the output variables from SDAE power flow models.
4. Simulation Results

In order to demonstrate the effectiveness of proposed SDAE power flow models and SPPF probabilistic power flow method, simulations are implemented on modified IEEE 39-bus system, modified IEEE 118-bus system and modified IEEE 300-bus system integrated with renewable energy.

4.1. System Information and Methods for Comparison

The system data of original IEEE 39-bus system, IEEE 118-bus system and IEEE 300-bus system is given in [24]. Three cases are modified by installing photovoltaic stations and wind farms at certain buses. For an illustrative purpose, tie-line power transfer is also modelled as wind farms or photovoltaic stations in a quite general view.

The choice of suggested parameters and model of wind farms and photovoltaic station for output powers as function of wind speed and solar radiation can be found in [25]-[28]. It is assumed that the power factors of wind farms’ generation were kept at 0.95 p.u. The load uncertainties are modeled by normal distributes. Five cases with different penetration rates and load variance are analyzed:

Case 1: Modified IEEE 39-bus system, and the maximum active power output of photovoltaic station is 200MW, rated active power output of wind farms is 260MW, and the standard deviations of load is 10%.

Case 2: Modified IEEE 118-bus system, and the maximum active output of photovoltaic station is 200MW, rated active power output of wind farms is 260MW, and the standard deviations of load is 10%.

Case 3: Modified IEEE 118-bus system, and the maximum active power output of the photovoltaic stations is 100MW at bus 13 and bus 16 and 150MW at bus 14 and bus 26. The rated active power output of wind farms at bus 59, bus 80 and bus 90 are 220MW, 200MW and 260MW respectively, and the standard deviations of load is 5%.

Case 4: Modified IEEE 118-bus system, 1% of the data is randomly lost among 50000 samples. The rest of parameters are the same as Case 2.

Case 5: Modified IEEE 300-bus system, and the forecasted active power outputs are 200MW for photovoltaic station and 260MW for wind farms. The standard deviation of load is 5%.

Methods M0-M4 are compared in calculating power flow and probabilistic power flow.

M0: M0 represents the MCS based on the Newton-Raphson algorithm, as Reference. Besides, parallel MCS performed on dual-core computer is also implemented for comparing time.

M1: Proposed SPPF method, with power flow calculated by SDAE power flow models, and there are 4 DAE structure in the SDAE, each of the DAE has 200 neurons.

M2: The MCS based on DC power flow methods.

M3: The MCS based on BP neural network (BPNN), and the number of neurons in hidden layer is 800, activation of BP neural network is sigmoid function.

M4: The MCS based on RBF neural network (RBFNN), the max number of neurons is 800, and the activation function is gauss function.

The end-criterion when training mentioned neural networks is that the change of the sum of weight matrix is below 1% for BP and SDAE, or neuron number for RBF neural network reaches maximum. The sample number of the MCS method is 50000.

4.2. Analysis of Results

4.2.1. Validity of power flow solvability judgment

With the Newton-Raphson (NR) algorithm taken as reference, this work assumes the power flow is unsolvable if the calculation is not converged after 50 iterations. In order to validate the accuracy of SDAE to judge the solvability of power flow, load level in Case 1 is enhanced and the judgment results are given in Table I. The correct rate means the number of correct judgement (the judged solvability is identical to actual solvability) over the number of total judgement.
Table 1. Judgement accuracy of power flow solvability with SDAE under different load levels in case 1

| Load Level | NR solvable / SDAE solvable | NR Solvable rates/ SDAE Solvable rates | Correct rate |
|------------|----------------------------|------------------------------------------|--------------|
| 100%       | 50000 / 50000            | 100.00%/100%                            | 100.00%      |
| 115%       | 49994 / 49992            | 99.9%/99.98%                            | 99.98%       |
| 125%       | 30888 / 30974            | 61.78%/61.95%                           | 99.59%       |

From Table I, we can see that the number of unsolvable samples is increasing with the enhancement of load level. When load level is 100%, 115% and 125%, the solvable rates of power flow obtained by NR methods is 100.00%, 99.99% and 61.78%, respectively; meanwhile, the solvable rates of power flow obtained by SDAE are 100.00%, 99.98% and 61.95%, respectively. The accuracy of solvability judgment is 100.00%, 99.99% and 99.59%, respectively. In conclusion, SDAE power flow models are able to judge the solvability of power flow with sound accuracy under various load level.

4.2.2. Accuracy of SDAE power flow models

The accuracy to calculate power flow together with the robustness against missing data and variation of system status using SDAE is verified in this section. The power flow and probabilistic power flow for Case 2, Case 3 and Case 4 are calculated using M0-M4 with the system status randomly sampled by the MCS. In order to compare the accuracy of different non-iterative methods in power flow calculation, the probability that error of active power flow exceeds 5MW ($P_{ef}$ in Table II) and probability that error of bus voltage exceeds 0.001 p.u. ($P_{ev}$ in Table II) is shown in Table II.

From Table II, while calculating the power flow for Case 2 and Case 3 with SDAE power flow models, the maximum and minimum $P_{ef}$ are 0.10% and 0.06%, and the maximum and minimum $P_{ev}$ are 0.05% and 0.03%. For other non-iterative methods, the maximum and minimum $P_{ef}$ are 22.61% and 11.17%, and their maximum and minimum $P_{ev}$ are 32.62% and 5.30%, and their errors are obviously higher than SDAE. In addition, from Case 4 when 1% of the input data is randomly lost, we can see that SDAE power flow models maintain a better accuracy compared with BP neural network and RBF neural network, while the Newton-Raphson algorithm and DC power flow models cannot directly calculate the power flow if the data is missing. In conclusion, the constructed SDAE power flow models demonstrate sound accuracy and robustness while calculating power flow.

Table 2. Accuracy comparison of power flow calculation with M0-M4

| Method | $P_{ef}$ | $P_{ev}$ |
|--------|----------|----------|
|        | Case 2   | Case 3   | Case 4   | Case 2   | Case 3   | Case 4   |
| M0     | 0        | 0        | -        | 0        | 0        | -        |
| M1     | 0.06%    | 0.10%    | 1.62%    | 0.05%    | 0.03%    | 0.48%    |
| M2     | 26.71%   | 22.61%   | -        | 74.98%   | 99.14%   | -        |
| M3     | 17.94%   | 20.93%   | 18.61%   | 32.62%   | 8.02%    | 32.76%   |
| M4     | 11.17%   | 15.23%   | 11.93%   | 26.41%   | 5.30%    | 26.73%   |

Note: “-” means the index is not available with specified method.

5. Conclusions

Uncertainties caused by power demand and renewable energy sources lead to the uncertainties of tie-line power transfer. Consequently, this paper presents an online probabilistic power flow algorithm based on SDAE and combined with the MCS. Firstly, benefiting from the approximation ability of SDAE in nonlinear power flow equations, SDAE power flow models are proposed. Two-stage training methods with cross-entropy loss function, mini-batch gradient descent algorithm and momentum learning rates are introduced to improve the parameters updating efficiency of SDAE. The well-trained
SDAE power flow models, performing high accuracy and robustness, are able to non-iteratively judge the solvability of power flow and calculate power flow. Besides, the proposed SPPF maintains accurate regardless of the penetration rates of renewable energy. Three standard IEEE test systems are simulated to verify the advantages of proposed SDAE power flow models and SPPF method. The proposed SPPF method, under equal condition, consistently outperforms the MCS that is based on other non-iterative methods including DC power flow, BPNN and RBFNN.

Appendix

Table A1. Nomenclature

| Symbol | Description |
|--------|-------------|
| $X$    | Input vector of DAE or SDAE |
| $\tilde{X}$ | Locally corrupted input vector from $X$ |
| $Y, Z$ | Middle layer vector and output vector of DAE |
| $W, b$ | Weight matrix and Biased matrix of SDAE or encoder in DAE |
| $W', b'$ | Weight matrix and biased matrix of decoder in DAE |
| $Y_t$ | Output vector of SDAE |
| $X_i, Y_i, Z_i$ | Input vector, middle layer vector and output vector of DAE in $l$th layer of SDAE |
| $w_{ij}^{(l,T)}$ | Weight from the $j$th neuron in middle layer of DAE in $l$-1th layer to the $i$th neuron in middle layer of DAE in layer $l$ after $T$th parameters updating |
| $b_i^{(l,T)}$ | Biased value of $i$th neuron of DAE in $l$th layer |
| $\eta$ | Learning rate of neural network |
| $r, m$ | Sequence number of initial sample and sample size in a mini-batch |
| $W^{(l,T)}, b^{(l,T)}$ | Weight matrix and biased vector of the DAE in $l$th layer of SDAE after $T$th parameters updating |
| $\Delta W^{(l,T)}, \Delta b^{(l,T)}$ | Change in weight matrix and biased vector of DAE in $l$th layer of SDAE after $T$th parameters updating |
| $dW^{(l,T-1)}, db^{(l,T-1)}$ | Change of $W^{(l,T-1)}$ compared with $W^{(l,T-2)}$ in $T$th parameters updating |
| $\rho$ | Momentum factor |
| $w_{ij}$ | Weight of $j$th neuron in middle layer towards $i$th neuron in output layer of top-layer DAE in SDAE |
| $Y_{i,l}$ | $i$th parameter in output vector of SDAE |
| $y_i$ | $i$th parameter in output of training sample |
| $y$ | Output vector of training sample |

Acknowledgments

This work was supported by Project supported by Science and Technology Project of State Grid Fujian Electric Power Company (2020ZXFXJ01).

References

[1] B. Borkowska, "Probabilistic load flow," IEEE Transactions on Power Apparatus and Systems, vol. PAS-93, no. 3, pp. 752-759, May 1974.
[2] R. N. Allan, B. Borkowska and C. H. Grigg, "Probabilistic analysis of power flows," Proceedings of the Institution of Electrical Engineers (London), vol. 121, no. 12, pp. 1551-1556, Dec. 1974.
[3] A. M. Leite da Silva, S. M. P. Ribeiro, V. L. Arienti, R. N. Allan and M. B. Do Coutto Filho, "Probabilistic load flow techniques applied to power system expansion planning," IEEE Transactions on Power Systems, vol. 5, no. 4, pp. 1047-1053, Nov. 1990.
[4] M. Fan, V. Vittal, G. T. Heydt and R. Ayyanar, "Probabilistic power flow analysis with generation dispatch including photovoltaic resources," IEEE Transactions on Power Systems, vol. 28, no. 2, pp. 1797-1805, May 2013.

[5] J. Yu and W. Dong, "Voltage security analysis on the distribution network integrated with wind power using probabilistic load flow," International Conference on E-Product E-Service and E-Entertainment, Henan, China, pp. 1-4, Nov. 2010.

[6] R. N. Allan and M. R. G. Al-Shakarchi, "Probabilistic techniques in a.c. load-flow analysis," Proceedings of the Institution of Electrical Engineers (London), vol. 124, no. 2, pp. 154-160, 1977.

[7] N. D. Hatzigiorgiou, T. S. Karakatsanis, and M. Papadopoulos, "Probabilistic load flow in distribution systems containing dispersed wind power generation," IEEE Transactions on Power Systems, vol. 8, no. 1, pp. 159-165, Feb. 1993.

[8] J. Tang, F. Ni, F. Ponci and A. Monti, "Dimension-adaptive sparse grid interpolation for uncertainty quantification in modern power systems: probabilistic power flow," IEEE Transactions on Power Systems, vol. 31, no. 2, pp. 907-919, Mar. 2016.

[9] C.-L. Su, "Probabilistic load-flow computation using point estimate method," IEEE Transactions Power Systems, vol. 20, no. 4, pp. 1843-1851, Nov. 2005.

[10] M. Aien, M. Fotuhi-Firuzabad and F. Aminifar, "Probabilistic load flow in correlated uncertain environment using unscented transformation," IEEE Transactions on Power Systems, vol. 27, no. 4, pp. 2233-2241, Nov. 2012.

[11] M. Aien, M. G. Khajeh, M. Rashidinejad and M. Fotuhi-Firuzabad, "Probabilistic power flow of correlated hybrid wind-photovoltaic power systems," IET Renewable Power Generation, vol. 8, no. 6, pp. 649-658, Aug. 2014.

[12] P. Chen, Z. Chen and B. Bak-Jensen, "Probabilistic load flow: A review," Third International Conference on Electric Utility Deregulation and Restructuring and Power Technologies, Nanjing, China, pp. 1586-1591, Apr. 2008.

[13] P. Wang, J. Zhang, Q. Gong, C. Tan, Y. Zhu and Y. Fan, "Reliability analysis of Bayesian Monte Carlo adaptive importance sampling method for structural safety," The Proceedings of 9th International Conference on Reliability, Maintainability and Safety, Guiyang, China, pp. 440-444, Jun. 2011.

[14] M. Hajian, W. D. Rosehart, and H. Zariepour, "Probabilistic power flow by Monte Carlo simulation with latin supercube sampling," IEEE Transactions on Power Systems, vol. 28, no. 2, pp. 1550-1559, May 2013.

[15] Z. Q. Xie, T. Y. Ji, M. S. Li and Q. H. Wu, "Quasi-Monte Carlo based probabilistic optimal power flow considering the correlation of wind speeds using copula function," IEEE Transactions on Power Systems, vol. 33, no. 2, pp. 2239-2247, Mar. 2018.

[16] S. P. Noopura, R. J. R. Kumar, A. Jain and M. V. Jayan, "Fast decoupled state estimation based on current equations," IEEE Innovative Smart Grid Technologies - Asia (ISGT ASIA), Bangkok, Thailand, pp. 1-6, Nov. 2015.

[17] A. Semlyen and F. de Leon, "Quasi-Newton power flow using partial Jacobian updates," IEEE Transactions on Power Systems, vol. 16, no. 3, pp. 332-339, Aug. 2001.

[18] B. Stott and O. Alsac, "Fast decoupled load flow," IEEE Transactions on Power Apparatus and Systems, vol. PAS-93, no. 3, pp. 859-869, May 1974.

[19] H. R. Baghaee, M. Mirsalim, G. B. Gharapetian and H. A. Talebi, "Fuzzy unscented transform for uncertainty quantification of correlated wind/PV microgrids: possibilistic-probabilistic power flow based on RBFNNs," IET Renewable Power Generation, vol. 11, no. 6, pp. 867-877, May 2017.

[20] K. Hornik, "Approximation capabilities of multilayer feedforward networks," Neural Networks, vol. 4, no. 2, pp. 251-257, Oct. 1990.

[21] N. L. Roux and Y. Bengio, "Representational power of restricted boltzmann machines and deep belief networks," Neural Computation, vol. 20, no. 6, pp. 1631-1649, Jun. 2008.
[22] J. Marques de Sá, L. Silva, J. Santos, and L. Alexandre, "Minimum error entropy classification," Springer-Verlag Berlin Heidelberg, Germany, vol. 420, 2013.

[23] P. Vincent, H. Latochelle, I. Lajoie and P. A. Manzagol, "Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion," The Journal of Machine Learning Research, vol. 11, no. 12, pp. 3371-3408, Mar. 2010.

[24] Power Systems Test Case Archive [Online]. Available: http://www.ee.washington.edu/research/pstca/pg_tca118bus.htm.

[25] Z. Qin, W. Li, X. Xiong, "Incorporating multiple correlations among wind speeds, photovoltaic powers and bus loads in composite system reliability evaluation," Applied Energy, vol. 110, no. 5, pp. 285-294, Oct. 2013.

[26] Y. Li, W. Li, W. Yan, J. Yu and X. Zhao, "Probabilistic Optimal Power Flow Considering Correlations of Wind Speeds Following Different Distributions," IEEE Transactions on Power Systems, vol. 29, no. 4, pp. 1847-1854, July 2014.

[27] J. Wang, M. Shahidehpour and Z. Li, "Security-constrained unit commitment with volatile wind power generation," IEEE Transactions on Power Systems, vol. 23, no. 3, pp. 1319-1327, Aug. 2008.

[28] M. Aien, M. Rashidinejad, and M. F. Firuz-Abad, “Probabilistic optimal power flow in correlated hybrid wind-PV power systems: A review and a new approach,” Renew. Sustain. Energy Rev., vol. 41, no. 4, pp. 1437–1446, Feb. 2015.