Discussion of Training Samples-Oriented Sampling Methods for Data-driven Power Systems Analysis

Jianglong Lei¹, Juan Yu¹, *, Yan Yang¹, Mingxu Xiang¹ and Hongxin Yu², Hongbin Wang²

¹The State Key Laboratory of Power Transmission Equipment & System Security and New Technology, College of Electrical Engineering, Chongqing University.
²State Grid Chongqing Electric Power Research Institute Chongqing, China, 400030.

*Corresponding author: yujuan@cqu.edu.cn

Abstract. Data-driven methods such as deep neural networks (DNNs) and Gaussian processes (GPs), have been a promising way to achieve the balance of the calculation accuracy and efficiency in power system analysis. For instance, studies are utilizing the DNN-based method to achieve the fast and accurate calculation of probabilistic power flow. These methods rely on the sample data for training, which is mainly obtained by sampling. However, there is still no guidance on how to select the suitable sampling method to effectively generate the representative training samples. To address this problem, this paper explores the impact of different sampling methods on data-driven methods through theoretical analysis and simulation. Then some reasonable conclusions are drawn in this paper: i) Random sampling (RS) is suitable for data-driven methods that require large-scale training samples. ii) Latin hypercube sampling (LHS) is suitable for data-driven methods that only work well under small-scale training samples. iii) The assumption that the training samples generated by uniform sampling (US), which covers all sampling intervals, could not lead to good performance. Finally, simulations are implemented on the IEEE 39-bus system and the results demonstrate the effectiveness of the proposed conclusions

1. Introduction

Data-driven methods such as deep neural networks (DNNs) [1] and Gaussian processes (GPs) [2] have been widely used in power system to addressed the problem of balancing calculation efficiency and accuracy. For these methods, training samples have a significant impact on their performance [3]- [4]. However, due to the privacy of power system data, it is difficult to collect the practical power system data. Hence, researchers usually use simulation methods to generate the required power system samples in many scenarios such as probabilistic power flow calculation [5]- [6].

The Monte Carlo Simulation (MCS) combined with random sampling (RS) is a typical simulation method [7]. In RS, the uncertain parameters $x$ is sampled according to its known probability density function $f(x)$. After sampling, the sample set $X$ that obeys this distribution can be generated. This method is easy to implement and has high accuracy. Therefore, a lot of studies choose RS to generate samples. Reference [8] utilizes RS to simulate the load fluctuations. In [9], the probability distribution characteristics affected by uncertain factors have been randomly sampled by RS.
However, to guarantee the calculation accuracy, a large number of samples are required by RS, which significantly increases the computational burden. Hence, the calculation efficiency of RS is relatively low and cannot be effectively applied in scenarios requiring high efficiency. To improve the calculation efficiency, the Latin Hypercube Sampling [10] (LHS), which belongs to stratified sampling, is proposed. This method effectively uses sampling values to reflect the overall distribution of random variables, and thus ensures that the sampling space can be fully covered by the sampling points. Hence, the number of required samples can be greatly reduced compared to RS. At present, many researchers have applied this method to power system analysis, such as power system adequacy planning [11], probabilistic power flow calculation [5]-[6], power system reliability analysis [12], etc. An LHS algorithm based on Gram-Schmidt sequence orthogonalization is proposed in [5], which effectively improves the sampling efficiency. A probabilistic power flow algorithm based on the extended LHS method is proposed in [6] for the drawback that the sampling number of the traditional LHS method must be fixed. Reference [12] combines the LHS method with the sequential MCS method and applies it to the reliability assessment of power generation systems containing renewable energy.

Because of the good performance of sampling methods in power systems analysis, they are also used by data-driven methods to generate the power systems training samples. Reference [3] utilizes the DNN-based method to solve the problem of probabilistic optimal power flow, which utilizes RS to generate the samples for training. In [13], a novel Gaussian process emulator is proposed, for the first time, to conduct the probabilistic power flow calculation, which has a good performance by using fewer samples generated by LHS. Also, uniform sampling (US) is used to generate samples in other areas, since it can cover all sampling intervals. For instance, reference [14] proposes the uniform local sampling method, which can sample local information from scene images more reasonably than RS for neural networks training. Different sampling methods have different characteristics, and they may be suitable for different scenarios. However, there are still no researches on how to choose effective sampling methods for different scenarios.

With these in mind, this paper explores the impact of different sampling methods on data-driven with theoretical and simulation analysis, and then draws some reasonable conclusions on how to generate the effective training samples, which are shown as follows. To the best of our knowledge, it is for the first time to discuss this topic for data-driven methods in power systems.

Conclusion 1. RS is suitable for data-driven methods that require large-scale training samples, such as DNNs.

Conclusion 2. LHS is suitable for data-driven methods that only work well under small-scale training samples, such as Gaussian processes.

Conclusion 3. The assumption that the training samples generated by US, which covers all sampling intervals, could not lead to good performance. Because the hypothesis that the training set and test set obey the independent and identical distribution (IID) cannot be guaranteed [15].

2. Outline of Sampling and Data-driven Methods

2.1. Random sampling
Random sampling (RS) is the basis of the MCS which is to randomly generate variables with known probability distributions. MCS combined RS can obtain high calculation accuracy when the sampling scale is large enough. It is often regarded as the benchmark method. However, due to the large-scale sampling, the calculation burden is too huge.

2.2. Latin Hypercube Sampling
LHS belongs to stratified sampling, which is an effective method to reflect the overall distribution of random variables with sampling values. Compared with RS, it has the following superiorities [5]: 1) Under the smaller-scale sampling, it is able to obtain the variables which are more consistent with their probability distribution; 2) The simulation results are more robust than RS, and multiple simulations results have little difference though LHS.
LHS contains two stages for sampling independent input random variables: sampling and sorting.

1) **Sampling:** The purpose of this stage is to enable sampling points to cover the whole probability space of variables. Details of the sampling stage are as follows:

- Let \( N \) be the sampling scale. \( X_1, X_2, \ldots, X_m \) are \( m \) random variables that need to be sampled, and \( P_i = F_i(X_i) \) is the cumulative probability distribution of \( X_i \), \( i = 1, 2, \ldots, m \);
- The range of \( P \) is \([0,1]\), and divide \( P \) into \( N \) equal intervals: \([1/N, 2/N], \ldots, [N-1/N, 1]\);
- Randomly sample a number from each interval as the sampling points, and then according to the inverse function of \( P \) to obtain the corresponding sample value of \( X_i \);
- Arrange the sampling value of each random variable in rows to form a sampling matrix.

2) **Sorting:** The purpose of this stage is to change the order of the sampled values of each variable to minimize the correlation of mutually independent random variables. The commonly used sorting methods in researches include Gram-Schmidt orthogonalization [5], Cholesky decomposition [16], and so on.

As illustrated above, LHS can ensure that the whole sampling space is covered by the sampling points, which is helpful for the analysis of the state with low probability. Hence, LHS can reflect the overall distribution of random variables with a smaller-scale sample.

### 2.3. Deep Neural Networks

There are different types of data-driven such as DNNs, GPs, and so on. Among them, DNNs have the deep structure and the strong function extracting capability, and thus they have been successfully applied in many areas. Stacked Denoising Auto-Encoders (SDAE) is a typical DNN, and its brief introduction is shown as follows.

The structure of SDAE is illustrated in Fig 1. With the encoding and decoding process, SDAE has strong capabilities of high-dimensional nonlinear feature extraction and high generalization capabilities, which has a wide range of applications in power systems [3] [4].
SDAE is stacked layer-by-layer by denoising auto-encoders (DAE). DAE has the same structure as the traditional Auto-Encoders (AE), except that the corruption process is added to the input, which is to make DAE reconstruct the pure input from the corrupted input. The structure of DAE is shown in Fig 2. DAE includes an input layer $\tilde{X}$, a corruption layer $X$, a middle layer $Y$, and an output layer $Z$. These layers are connected through encoding and decoding process. The purpose of training DAE is to extract the input feature through the target expression. Therefore, the output is set as the input itself during training. The detailed calculation process is as follows. The corruption process is shown in (1).

$$X \sim C_D(\tilde{X})$$

where $\tilde{X}$ is the input, and $C_D$ is the corruption process, $X$ is the input after corruption. Then the middle layer $Y$ can be obtained through the encoding function, as shown in (2).

$$Y = f_\theta(X) = a(WX + b)$$

where $Y$ is the middle layer. $f_\theta$ is the encoding function, which extracts the relationship between $Y$ and $X$. $a(x)$ is the activation function, and the rectified linear unit (ReLU) is selected as the activation function in the encoding process: $a(x) = \max(0, x)$. The weight matrix $W$ and biased vector $b$ are parameters in encoding process.

Finally, the decoded output $Z$ can be obtained through the decoding function, as shown in (3).

$$Z = g_\sigma(Y) = a(W'Y + b')$$

where $g_\sigma$ is the decoding function, which extracts the relationship between $Y$ and $Z$. $a(x)$ is the activation function, and the linear function is selected as the activation function in the decoding process: $a(x) = x$. The weight matrix $W'$ and biased vector $b'$ are parameters in decoding process.

As shown in Fig. 1, SDAE is stacked layer-by-layer by the encoding process of DAE. Remarkably, the output layer of DAE is not involved in the data flow of SDAE. And SDAE extracts related future by continuously encoding the input layer $\tilde{X}$. From the structure diagram, we can deduce the formula of the output $Y$ of the SDAE.
The specific introduction of SDAE can be found in [4].

2.4. Gaussian processes

Gaussian process is a kind of stochastic process in probability theory and mathematical statistics. It is a combination of a series of random variables that obey a normal distribution in an index set. Modeling and predicting Gaussian processes is an important part of data-driven methods, signal processing and other areas. Common models of GP include Gaussian Process Regression (GPR) and Gaussian Process Classification (GPC). GPR is taken as an example to introduce GP.

Given N sets of learning samples $x = [x_1, x_2, \ldots, x_m]$, $y = [y_1, y_2, \ldots, y_m]$, $x_*$ is the test samples and $f_*^i$ is the probability distribution of test results. The model of Gaussian process regression can be established by (5).

$$f(x) = X^T \omega, \quad y = f(x) + \epsilon$$

where $\omega$ is the weight coefficient and $\epsilon$ is the residual or noise.

GPR takes the prior of the function space $f(x)$ as the Gaussian process, as shown in (6):

$$f(x) \sim GP(\mu(x), k(x,x'))$$

where $\mu(x)$ represents the mean function, which returns the mean of each dimension, $k(x)$ is the covariance function or kernel function, which returns the covariance matrix between the dimensions of two vectors. Gaussian process can be uniquely defined by mean function and covariance function.

The probability distribution of $x_*$ and $f_*$ is represented by (7).

$$y \sim N(0, K + \sigma^2 I), \quad f_* \sim N(0,k(x_,x_))$$

where $K$ represents the Gram matrix obtained from the learning sample, $\sigma$ is the covariance of Gaussian distribution, $k$ is the kernel function.

The joint probability distribution is shown in (8):

$$\begin{bmatrix} y \\ f_* \end{bmatrix} \sim N \left( \begin{bmatrix} 0 \\ 0 \end{bmatrix} \begin{bmatrix} K(x,x) + \sigma^2 I & K(x,x_*) \\ K(x_*,x) & k(x_,x_*) \end{bmatrix} \right)$$

The GPR equation formula can be obtained by the marginalization property of the joint normal distribution. The predicted result $\tilde{f}_*$ of the GPR model can be expressed as follows.

$$p(f_* | x, y, x_*, \sigma^2) = N \left[ f_* | \bar{f}_*, \text{cov}(f_*) \right]$$

$$\tilde{f}_* = k(x_*,x)(K + \sigma^2 I)^{-1} y$$

$$\text{cov}(f_*) = k(x_*,x) - k(x_*,x)(K + \sigma^2 I)^{-1} k(x_,x)$$
3. Sampling Methods Analysis in Data-driven

This section theoretically analyzes the application scenarios of different sampling methods for data-driven methods.

Conclusion 1: RS is suitable for data-driven methods that require large-scale samples. With the large enough sampling scale, RS is spatially optimal based on the information theory [17]. Hence, as long as the sampling scale is sufficient, RS can better reflect the distribution characteristics of random variables than other sampling methods. Hence, we should not take it for granted that LHS always outperforms RS, and RS is more suitable for the scenarios that the requirement of sampling scale is large, such as DNNs, which require a large number of training samples because of their complex structure.

Conclusion 2: LHS is suitable for data-driven methods that only work well under small-scale training samples. Compared with RS, LHS can cover the entire sampling space more effectively, especially for the edge of the sampling space. Hence, the distribution of random variables can be well obtained through a smaller sampling scale. GPR is a non-parameter model. Because its computational complexity increases as the amount of data, it is only suitable for processing small samples. Hence, LHS is suitable to generate samples for this method than RS.

Conclusion 3: The assumption that the training samples generated by US, which covers all sampling intervals, could not lead to good performance. In theory, if the training sample can fully cover the sampling space, the training set can contain more data information, and thus enable the data-driven methods to extract the data features more effectively during training. US is the one that can cover the whole sampling space to some extent. US assumes that the probability of random variables is equal for each possible sampling interval. Hence, the events with low probability can be sampled by this sampling method. However, the events with large probability would be relatively less sampled under the same sampling scale. Also, the actual distribution of random variables would be changed by US, which could result in bad training performance. Therefore, US is not suggested to generate training samples for data-driven methods, as these methods are based on the independent and identical distribution hypothesis of the training set and test set.

4. Numerical Test

Take probabilistic power flow as an example, case studies are conducted on the modified IEEE 39-bus system [18] with MATPOWER package using MATLAB® R2020a version. Three wind farms and two photovoltaic stations are connected to the modified system. The uncertain inputs include the fluctuation of load, wind velocity, and solar insolation. In this paper, it is assumed that the load follows the normal distribution with 10% standard deviations. Wind velocity is assumed to follow the Weibull distribution with the shape parameter of 2.06 and the scale parameter of 7.17. The capacity of each wind farm is 50MW. Solar insolation is assumed to follow a Beta distribution with shape parameters of 2.06 and 2.5. The capacity of each solar insolation is 100MW.

4.1. System Information

To prove the validity of the conclusions drawn in section III, different cases are implemented to compare the performance of different sampling methods. The compared sampling methods (M0-M2) are listed in Table I. The information about the cases is shown as follows.

Case 1: The selected data-driven method is DNN.
Case 2: The selected data-driven method is GPR.

In Case 1, the structure of DNN has two middle layers and each layer has 200 neurons. In Case 2, the squared exponential kernel is selected as the kernel function of GPR.

All cases in this paper are tested on a PC with AMD Ryzen 5 3600X 6-Core Processor 3.80 GHz, 16 GB RAM.

The following indices are used to judge the different methods:

\[ P_{V, \rho} \]: Probability that the absolute error of voltage amplitude exceeds \( \rho \) p. u.
\( P_{v, \rho} \): Probability that the absolute error of voltage angle exceeds \( \rho \) rad.
\( P_{P, \rho} \): Probability that the absolute error of active power flow exceeds \( \rho \) MW.
\( P_{Q, \rho} \): Probability that the absolute error of reactive power flow exceeds \( \rho \) MVar.

**Table I.** Compared Methods in This Section.

| Method | Description            |
|--------|------------------------|
| M0     | MCS combined with RS   |
| M1     | MCS combined with LHS  |
| M2     | MCS combined with US   |

4.2. Analysis of Results

4.2.1. Performance of sampling methods based on DNN. To prove the effectiveness of Conclusion 1, the following indices are shown in Table II to compare the accuracy of the probabilistic power-flow calculation under different sampling methods: \( P_{V,0.002} \), \( P_{\theta,0.003} \), \( P_{P,4} \), \( P_{Q,0.8} \). In this case, the test set for M0-M2 is sampled by RS, which is in accordance with the practical scenario. The number of test samples is 10000, which is large enough to reflect the distribution of random variables in power systems. The indices under the small-scale (3000 samples, denoted as S) and large-scale (10000 samples, denoted as L) training samples are listed in Table II, respectively.

**Table II.** Performance of sampling methods based on DNN.

| Method | \( P_{V,0.002} \) (%) | \( P_{\theta,0.003} \) (%) | \( P_{P,4} \) (%) | \( P_{Q,0.8} \) (%) |
|--------|------------------------|------------------------|-----------------|------------------|
|        | S          | L          | S               | L               |
| M0     | 23.2       | 5.59       | 43.5            | 3.59             |
| M1     | 23.5       | 5.77       | 45.0            | 4.89             |
| M2     | 38.0       | 29.6       | 61.7            | 27.3             |

It can be observed from Table II that under the small-scale training samples, the error indices of all methods are high. This is because the structure of DNN is complex and it requires a large number of samples for training. Under the large-scale training samples, it can be observed that all indices of M0 are better than M1 and M2. It is demonstrated that compared with other sampling methods, RS is more suitable for data-driven methods that require large-scale samples.

Moreover, the performance of M2 needs to be further illustrated combining the result of the following simulation.

4.2.2. Performance of sampling methods based on GPR. To verify that LHS is suitable for data-driven methods that can only handle the small-scale training samples (Conclusion 2), this subsection compares the training results of different sampling methods under small-scale training samples based on GPR. Firstly, the CPU times consumed by GPR under different sampling scales are illustrated in Table III. Then, the following indices are shown in Table IV: \( P_{V,0.001} \), \( P_{\theta,0.006} \), \( P_{P,4} \), \( P_{Q,0.3} \). In TABLE IV., the test set for M0-M2 is also sampled by RS. In this case, the number of test samples is 10000. The indices under 300 training samples are listed in Table IV.

**Table III.** CPU Times of Different Sampling Scales in GPR.

| Method | CPU Times |
|--------|-----------|
|        | 300 samples | 3000 samples |
| M1     | 39 s       | 3195 s       |
It can be seen from TABLE III that the CPU times of GPR increases explosively as the number of training samples. Hence, GPR is only suitable for processing the small-scale training samples.

**TABLE IV. Performance of sampling methods based on GPR.**

| Method | $P_{V,0.001}$(%) | $P_{u,0.006}$(%) | $P_{P,0.4}$(%) | $P_{Q,3}$(%) |
|--------|----------------|----------------|----------------|--------------|
| M0     | 5.21           | 1.48           | 1.64           | 3.04         |
| M1     | 4.70           | 1.42           | 1.50           | 2.70         |
| M2     | 6.95           | 6.38           | 7.91           | 4.18         |

It can be observed from Table IV that all indices of M1 are better than M0 and M2. These results prove that LHS has a better performance than the other two methods with small-scale training samples. Moreover, it can be seen that all indices of M2 are worst in Table II and Table IV, which proves Conclusion 3 that US does not suitable for the data-driven methods. Because the training set and test set do not obey the IID hypothesis under this sampling method.

Therefore, the effectiveness of the proposed conclusions has been verified by the above simulation results.

5. **Conclusion**

This paper explores the impact of different sampling methods on data-driven through theoretical analysis and simulation. We find that sampling methods cover the whole sample spaces may not the best choice, which needs to be selected according to the specific application scenario. Moreover, it is more important to ensure that the training set and test set obey the IID hypothesis. Simulations are performed on the modified IEEE 39-bus system. The simulation results have demonstrated the effectiveness of the proposed conclusions.

**Acknowledgment**

The authors are grateful for the support from Science and Technology Project of State Grid Corporation of China (5100-201999333A-0-0-00).

**References**

[1] Schmidhuber, Jürgen. Deep Learning in Neural Networks: An Overview[J]. Neural Netw, 2015, 61:85-117.

[2] Seeger M. Gaussian processes for machine learning[J]. International journal of neural systems, 2004, 14(2):69.

[3] Y. Yang, Z. Yang, J. Yu, B. Zhang, Y. Zhang and H. Yu, "Fast Calculation of Probabilistic Power Flow: A Model-Based Deep Learning Approach," in IEEE Transactions on Smart Grid, vol. 11, no. 3, pp. 2235-2244, May 2020, doi: 10.1109/TSG.2019.2950115.

[4] Xiang M, Yu J, Yang Z, et al. Probabilistic power flow with topology changes based on deep neural network[J]. International Journal of Electrical Power & Energy Systems, 2020, 117:105650-.

[5] YU Han, CHUNG Chiyong, WONG Kitpo, et al. A probabilistic load flow calculation method with Latin hypercube sampling[J]. Automation of Electric Power Systems, 2009, 33(21):32-35.

[6] DING Ming, WANG Jingjing, LI Shenghu. Probabilistic load flow evaluation with extended Latin hypercube sampling[J]. Proceedings of the CSEE, 2013, 33(4):163-170.

[7] HASTINGS, W. K. Monte Carlo sampling methods using Markov chains and their applications[J]. 1970, 57(1):97-109.

[8] WU Hong, ZHAO Xia, ZHOU Jiaqi. Monte Carlo Simulation of Probabilistic Small Disturbance Stability[J]. Automation of Electric Power Systems, 2009, 33(11):8-12.
[9] ZHAO Yuan, GUO Yin, XIE Kaigui. Research on Probability Distribution Characteristics of Bulk Power System Reliability Considering Parameter Uncertainty[J]. Power System Technology, 2013, (8):2165-2172.

[10] M.D. Mckay, Beckkman R J, Conover W J. Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code[J]. Technometrics, 1979, 21(2):266-294.

[11] Jirutitijaroen P, Singh C. Reliability Constrained Multi-Area Adequacy Planning Using Stochastic Programming With Sample-Average Approximations[J]. IEEE Transactions on Power Systems, 2008, 23(2):504-513.

[12] SHU Z, JIRUTITIJAROEN P. Latin hypercube sampling techniques for power systems reliability analysis with renewable energy sources[J]. IEEE Trans on Power Systems, 2011, 26(4):2066-2073.

[13] Xu Y, Hu Z, Mili L, et al. Probabilistic Power-Flow Calculation Based on a Novel Gaussian Process Emulator[J]. IEEE Transactions on Power Systems, 2020, PP(99):1-1.

[14] Zhan Chunru. Research of scene classification based on convolutional neural networks[D]. 2018.

[15] Pan S J, Yang Q. A Survey on Transfer Learning[J]. IEEE Transactions on Knowledge & Data Engineering, 2010, 22(10):1345-1359.

[16] Yu H, Chung C, Wong K P, et al. Probabilistic Load Flow Evaluation With Hybrid Latin Hypercube Sampling and Cholesky Decomposition[J]. IEEE Transactions on Power Systems, 2009, 24(2):661-667.

[17] Acharya, Swarup, Gibbons, et al. Congressional samples for approximate answering of group-by queries[J]. ACM SIGMOD Record, 2000.

[18] Zimmerman R D, Murillo-Sanchez C E, Thomas R J. MATPOWER: Steady-State Operations, Planning, and Analysis Tools for Power Systems Research and Education[J]. IEEE Transactions on Power Systems, 2011, 26(1):12-19.