A Parallel Multi-Modal LSTM for Power System Model Calibration

This paper was downloaded from TechRxiv (<https://www.techrxiv.org>).

LICENSE

CC BY 4.0

SUBMISSION DATE / POSTED DATE

28-01-2022 / 03-02-2022

CITATION

khazeiynasab, Seyyed Rashid (2022): A Parallel Multi-Modal LSTM for Power System Model Calibration. TechRxiv. Preprint. <https://doi.org/10.36227/techrxiv.19090031.v1>

DOI

[10.36227/techrxiv.19090031.v1](https://doi.org/10.36227/techrxiv.19090031.v1)

A Parallel Multi-Modal LSTM for Power System Model Calibration

Seyyed Rashid Khazeinyasab
Electrical and Computer Engineering
University of Central Florida
rashid@knights.ucf.edu

Abstract—Renewable energies, smart loads, energy storage, and new market behavior are adding new sources of uncertainty to power systems. Therefore, planning in real-time and developing high-quality models is crucial to adapt to uncertainties. Model validation based on actual measurements is necessary for obtaining accurate representations of power systems dynamics with system uncertainties. This paper presents a new measurement-based method to calibrate the parameters of a synchronous generator by deep learning method based on the long-short term memory (LSTM) network. First, critical parameters are determined regarding the active/reactive behavior of the generator. Then, a parallel multimodal LSTM (PM-LSTM) is designed with flexible input time steps to capture important features of temporal patterns from time-series measurements. The extracted features are then fed into a dense layer to capture the joint representation of inputs. The simulations conducted for a hydro generator under different events show that the proposed method can estimate the model parameters accurately and efficiently.

Index Terms—Deep learning, long short-term memory, generator model, parameter calibration, PMU, synchrophasor.

I. INTRODUCTION

Power system monitoring, protection, and control are primarily model-based. Therefore, an accurate dynamic model for power system models is essential for a proper evaluation of the dynamic behavior of the system. Inaccurate parameter values in the dynamic models may lead to inappropriate control actions, resulting ultimately in power system failures.

For traditional parameter calibration methods (offline methods or field tests), the generator has to be out of service, which could be time-consuming and labor-intensive and is not economically viable [1]. Online methods based on synchrophasor measurements have been proposed to overcome the limitations of the offline techniques. Similarly, The method based on black-box optimization [2], is efficient for low-dimensional parameter cases but fails to estimate the parameters accurately for high-dimensional cases. Gradient-based optimization method have been proposed in [3] for generator parameter estimation, but have to calculate the gradient of a nonlinear objective function with respect to the parameters in the generator dynamic model in every iteration and may get stuck at a local optima.

Kalman filter-based methods have also been proposed for generator model calibration [4], [5]. However, these methods usually suffer from slow convergence rate. Bayesian inference methods have also been proposed for generator parameter estimation [6], [7]. The proposed method in [6] uses a surrogate model instead of the likelihood function, which can only estimate the parameters for a model with a low number of

parameters. Besides, the performance of the method in [7] may deteriorate for high dimensional parameters. The proposed adaptive approximate Bayesian computation with sequential Monte Carlo sampler (A-ABC-SMC) method in [8] has a good performance under high dimensional. However, if the parameter space is large, the algorithm can not accurately estimate the parameters.

With the rapid development of the learning method, it is possible to use learning techniques to represent the complicated nonlinear relationship of the outputs of the generator with its parameters [9], [10]. Recently, machine learning approaches have been applied for parameter calibration. The proposed method in [11] generates extensive simulation data to train a multi-output convolutional neural network to estimate the small number of generator parameters. The proposed q-learning method [12], deep q-learning-based method [13], and soft actor-critic deep learning-based method [14] are extremely dependent on the hyper parameters and reward functions. It is challenging, especially for high-dimensional cases, to design a suitable reward function, to find the global optimum as the estimated parameters.

Long-short term memory (LSTM)-based methods are powerful techniques for representing complex nonlinear relationships. To calculate the generator model parameters more efficiently and more accurately, the LSTM technique is developed. The PMU measurement time series are captured with powerful temporal patterns by means of an LSTM network with a flexible number of temporal states. Further, the extracted features are fed into a linear regression model to estimate the full probability distributions of the generator parameters.

This paper proposes a generator parameter calibration approach based on LSTM and present a parallel multimodal LSTM (PM-LSTM) to further improve the efficiency. Compared to the existing methods, the advantages and improvements of the proposed method are as 1) It can provide the full distributions of the parameters even when the generator model has significant model discrepancies due to gross errors in the parameters. 2) It can estimate the high dimensional cases accurately even when the parameter space is large enough. 3) The proposed method is simulation-based and does not require a likelihood function or state-space model of the generator. Since commercial software already has stability models, its implementation is much easier and can be used very easily in real-world applications.

The remainder of this paper is organized as follows. Section II presents the problem statement and introduces the proposed

framework. Section III introduces the sensitivity approach for identifying critical parameters. In Section IV, a detailed overview is provided for the deep learning-based method, and a PM-LSTM based method is developed for generator parameter estimation. Section V presents case studies to validate the proposed method. Finally, conclusions and future works are given in Section VI.

II. GENERATOR DYNAMIC MODEL AND EVENT PLAYBACK

The nonlinear dynamical model of a synchronous generator and its controls includes a generator (GENTPJ), an exciter (ESST1A), a governor (IEEEG3), and a power system stabilizer (PSS, IEEEEST), and can be written in a general form as:

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, \boldsymbol{\alpha}) \\ \mathbf{y} = \mathbf{g}(\mathbf{z}, \mathbf{u}, \boldsymbol{\alpha}), \end{cases} \quad (1a) \quad (1b)$$

where $\mathbf{f}(\cdot)$ and $\mathbf{g}(\cdot)$ are state transition and output functions, $\mathbf{x} \in \mathbb{R}^n$ is the state vector that includes rotor angle, rotor speed, transient or sub-transient voltages, and the controller states, $\mathbf{u} = [V \ \theta]^\top$ is the injected signal vector of voltage magnitude V and phase angle θ , $\mathbf{y} \in \mathbb{R}^o$ is the output vector for active power P and reactive power Q of the generator, and $\boldsymbol{\alpha} \in \mathbb{R}^v$ is the parameter vector. The details of the models of generator, exciter, governor, and PSS are can be found in [1], [8].

In model validation, the time-series real and reactive powers from the model, denoted by vectors $\mathbf{P}_m$ and $\mathbf{Q}_m$, are compared with the time-series real and reactive power measurement vectors, denoted by $\mathbf{P}_{\text{meas}}$ and $\mathbf{Q}_{\text{meas}}$. If the outputs from the model do not match the PMU-measured outputs, the generator parameters need to be calibrated [5]. Let $\mathbf{z}^* = [\mathbf{P}_{\text{meas}} \ \mathbf{Q}_{\text{meas}}]^\top$ be the measurements from PMU and $\mathbf{z} = [\mathbf{P}_m \ \mathbf{Q}_m]^\top \in \mathbb{R}^{2K}$ be the time-series outputs of model (1) where K is the number of time steps.

In the event playback, a PMU is required to be installed at the subsystem point of common coupling (PCC) to collect measurements of voltage magnitude and phase angle/frequency, and current phasors of the branches connected to the PCC. Real and reactive power measurements can be obtained from the voltage phasor and current phasor measurements. An event must be triggered to collect the model calibration dataset, such as tripping a branch or inserting a fault in the test system. Thus, disturbance dynamics can be obtained using PMU at the point of common coupling of the subsystem. If a large model deficiency is found between model simulation and PMU measurements for an event captured by PMU, then the parameters need to be calibrated.

III. IDENTIFYING CRITICAL PARAMETERS

After a model deficiency has been revealed, the next step is to identify the problematic parameters. A generating unit with its control can have many parameters. Calibrating all parameters could be computationally challenging, and also, not every parameter is identifiable [15]. Specifically, the sensitivity

of the outputs P_m and Q_m with regard to parameter α_i can be calculated as [5]:

$$S_P(\alpha_i) = \sum_{k=1}^K \frac{|P_{m,k}(\alpha_i^+) - P_{m,k}(\alpha_i^-)|}{K(\alpha_i^+ - \alpha_i^-)/\alpha_i},$$

$$S_Q(\alpha_i) = \sum_{k=1}^K \frac{|Q_{m,k}(\alpha_i^+) - Q_{m,k}(\alpha_i^-)|}{K(\alpha_i^+ - \alpha_i^-)/\alpha_i},$$

where K is the number of time steps, $\alpha_i^+ = \alpha_i + \Delta\alpha_i$ and $\alpha_i^- = \alpha_i - \Delta\alpha_i$, and $\Delta\alpha_i$ is a small perturbation of α_i . After the sensitivity analysis, the parameters selected to be estimated are those with a large sensitivity [5], [15].

IV. LSTM BASED PARAMETER ESTIMATION

Generator parameter estimation based on PMUs is a mathematical representation of the highly nonlinear relationship between parameters and outputs. Based on the generator model in Section II, one can define the complex nonlinear relationship of the generator model parameters and the measurements using the generalized recursion computed by

$$\boldsymbol{\alpha}_c^t = \mathcal{F}(\mathcal{F}_P(\rho^{t-1}, \dots, \rho^{t-t_p}), \mathcal{F}_Q(\sigma^{t-1}, \dots, \sigma^{t-t_q})), \quad (2)$$

where, ρ is the $t_p \times \eta$ -dimensional vector of real power features, and σ is the $t_q \times \zeta$ -dimensional vector of reactive power features. As shown in (2), the generator parameters at time step t is a nonlinear function $\mathcal{F}$ of the temporal features extracted by $\mathcal{F}_P$ from active power and $\mathcal{F}_Q$ from reactive power. Generator parameter estimation is to estimate the $\boldsymbol{\alpha}_c^t$ by learning the optimal $\mathcal{F}$ [16], [17]. However, this function can not be formulated analytically, and the generator parameters can not be estimated. By incorporating a hidden layer of memory cells into the recurrent neural network, LSTM can establish the temporal correlation between previous information and the current conditions [18]. This property can help estimate the parameters of the generator from its captured dynamics. LSTM has three special multiplicative computational units named as the input, output, and forget gate. The input gate is responsible for controlling the portion of the current state information, and the current input should be saved to the memory cells [19]. The forget gate controls the value of the last time cell, which will be saved, and the output gate controls the output of the cell. The structure of the LSTM network is shown in Fig. 1. The formulations of the different gates in an LSTM will be discussed below. The forget gate is as follow:

$$\varrho_t = \kappa(W_f * [h_{t-1}, x_t] + b_f),$$

where h_{t-1} is the short-memory state transmitted by the previous LSTM unit, x_t is the input, ϱ_t is the forget gate, and $\kappa(\cdot)$ is a nonlinear smooth and differentiable activation function and usually considered as a *sigmoid* function, whose output is a value between zero and one. Which, zero means “let nothing pass” and one means “let everything pass” [20]. The forget gate output tells the cell state which information to

Fig. 1. Structure of the LSTM block.

forget by multiplying zero or one. The input gate controlled by the following equation

$$I_t = \kappa(W_i * [h_{t-1}, x_t] + b_i),$$

where W_i and b_i are the weight and the bias of the input gate, respectively. The new candidate value $\tilde{c}_t$ and the future cell state c_t are obtained by the following equations

$$\begin{aligned} \tilde{c}_t &= \tilde{\kappa}(W_c * [h_{t-1}, x_t] + b_c), \\ c_t &= q_t \circ c_{t-1} + I_t \circ \tilde{c}_t, \end{aligned}$$

where W_c , and b_c are the weight and the bias of the candidate layer, and $\tilde{\kappa}(\cdot)$ is a $\tanh$ function for the memory content update value. The output gate unit is updated as following

$$o_t = \kappa(W_o * [h_{t-1}, x_t] + b_o),$$

where W_o and b_o are the weight and bias of the output layer. The final output of short term memory h_t is obtained by the following equation

$$h_t = o_t \circ \tilde{\kappa}(c_t).$$

Where $[\cdot, \cdot]$ is the concatenation operator that merges two tensors, while $*$ is the matrix multiplication operator and $\circ$ is the Hadamard (element-wise) product [18]. The latent variable vector obtained from the LSTM at the t th round of network update is denoted by h_t , which is the t th element of the parameter temporal feature tensor h . The full set of the LSTM parameters is updated by performing stochastic gradient descent at each time step t .

Let real and reactive power of the model for $\alpha_{c,i}$ be $\mathbf{P}_{m,i} = [p^1, p^2, \dots, p^K]^\top$, and $\mathbf{Q}_{m,i} = [q^1, q^2, \dots, q^K]^\top$. To use the LSTM, the inputs and outputs should be the time series data. Then, in parameter estimation, a vector of parameters should be provided. For this reason, the following vector is constructed $\alpha_{c,i}$, with a size of K . Since the parameters of the generator are constant during the estimation, in this paper, it is assume that for $z_i(\alpha_{c,i})$, the parameter vectors is $\alpha_{c,i} = [\alpha_{c,i}^1, \alpha_{c,i}^2, \dots, \alpha_{c,i}^K]^\top$, and the values of that are constant, i.e. $\alpha_{c,i}^1 = \alpha_{c,i}^2 = \dots = \alpha_{c,i}^K$. A time-series parameter vector for the model under study, in addition to the

time series of real power and reactive power, can be used to estimate its parameters.

Based on Section III, some parameters have more impacts on the real power, denoted by α_c^p , and some of them have more effects on the reactive power, denoted α_c^q . With having N dynamic simulations, let define $\mathbf{P}_m$ which contains N model real power, i.e. $\mathbf{P}_m = [\mathbf{P}_{m,1}, \mathbf{P}_{m,2}, \dots, \mathbf{P}_{m,N}]^\top$, and $\mathbf{Q}_m$ which contains N model reactive power, i.e. $\mathbf{Q}_m = [\mathbf{Q}_{m,1}, \mathbf{Q}_{m,2}, \dots, \mathbf{Q}_{m,N}]^\top$, and the parameter as $\Theta_c = [\alpha_{c,1}, \alpha_{c,2}, \dots, \alpha_{c,N}]^\top$. Based on the engineering experience [12], the steady-state mismatch can be determined strongly by the turbine governor model. Some parameters such as T_6 and T_5 only shift the total waveform of the generator outputs. Then, in this paper, another feature extracted from the PMU measurements is used, which only reflects the impacts of these parameters to help the algorithm identify the parameters more accurately. This feature is derived as follows. Let define $\mathbf{P}_{md,i} = [p^1, p^2 - p^1, p^3 - p^2, \dots, p^K - p^{K-1}]^\top$, and $\mathbf{Q}_{md,i} = [q^1, q^2 - q^1, q^3 - q^2, \dots, q^K - q^{K-1}]^\top$, then two additional features, $\mathbf{P}_{md}$ and $\mathbf{Q}_{md}$ are used as the inputs for training the model which are defined as follows. $\mathbf{P}_{md} = [\mathbf{P}_{md,1}, \mathbf{P}_{md,2}, \dots, \mathbf{P}_{md,N}]^\top$, $\mathbf{Q}_{md} = [\mathbf{Q}_{md,1}, \mathbf{Q}_{md,2}, \dots, \mathbf{Q}_{md,N}]^\top$. Two parallel multimodal LSTMs (M-LSTM) were built to build a comprehensive model. One M-LSTM is for active power, LSTM-A, and one M-LSTM for reactive power, LSTM-R. The outputs of the LSTM-A are the parameters of the system that have more effects on the active power, whereas those of the LSTM-R are the parameters that have more effects on the reactive power.

The inputs tensors for LSTM-A considered as $\mathcal{P}_A = [\mathbf{P}_m, \mathbf{P}_{md}]^\top \in \mathbb{R}^{N \times K}$, and $\mathcal{Q}_R = [\mathbf{Q}_m, \mathbf{Q}_{md}]^\top \in \mathbb{R}^{N \times K}$ as inputs for LSTM-R. The maximum lag values for the time series active and reactive power are considered as t_p and t_q . At each training time step t , LSTM-A observes $\rho_{t_p \times \eta}$ which is from $\mathcal{P}_A$ and includes ρ_{t_p} to ρ_1 ; and i th LSTM input is a tensor with size $1 \times \eta$ and the corresponding output is h_i^p . Similarly, LSTM-R observes $\sigma_{t_q \times \zeta}$ from $\mathcal{Q}_R$ that includes σ_{t_q} to σ_1 and i th LSTM input is a tensor with size $1 \times \zeta$ and the corresponding output is h_i^q . When all the input vectors in LSTM-A and LSTM-R are observed, the average temporal latent variables at time step t is computed as follows [16]:

$$\begin{aligned} \psi_t^p &= \frac{1}{t_p} \sum_{k=0}^{t_p} h_k^p, \\ \psi_t^q &= \frac{1}{t_q} \sum_{k=0}^{t_q} h_k^q, \end{aligned}$$

where h_k^p and h_k^q are the latent variable vectors for active/reactive power obtained from the LSTM-A and LSTM-R at the k th round of network update. At each time step t , the temporal averages of the parameters set are fed into the linear regression models, $\phi_\rho(\cdot)$ and $\phi_\zeta(\cdot)$, and the parameters at t are estimated. The structure of PM-LSTM is shown in Fig. 2. Note that in the PM-LSTM model, the number of LSTM layers in LSTM-A is t_p , and the number of LSTM layers in

Fig. 2. The proposed PM-LSTM for generator parameter estimation.

LSTM-R is t_q .

V. SIMULATION RESULTS

The proposed method is tested based on power system models, which are implemented in PSS/E software. The same hydro-generator model and PMU data in [1], [8] are used. Two metrics are used to evaluate the performance of different methods, including: mean absolute error (MAE), and mean squared error (MSE).

A. Critical Parameter Identification

For sensitivity analysis, a small perturbation $\Delta\alpha_i = 5\%|\alpha_i|$ is applied to each parameter. From the trajectory sensitivity, it is found that only the parameters shown in Table I are the problematic parameters for the system among all the synchronous generator parameters. The top eighteen critical parameters and their active and reactive power sensitivities are listed in Table I. Based on Table I, for the eighteen parameters case, the parameter set $\{T_6, K_s, T_5, K_A, T_b, T'_{do}, T_3, T_1, X_d\}$ is the output of LSTM-R, and $\{a_{23}, H, T_c, R_T, X'_d, X'_q, T_R, R_P, A_1\}$ set is estimated with LSTM-A.

TABLE I
TOP EIGHTEEN CRITICAL PARAMETERS OF THE MODEL

Param.	$S_P(\alpha_i)$	$S_Q(\alpha_i)$	Param.	$S_P(\alpha_i)$	$S_Q(\alpha_i)$
T_6	0.13	26.26	X_d	0.02	0.70
K_s	0.11	26.25	T_c	0.60	0.20
T_5	0.12	22.11	H	0.58	0.05
K_A	0.1	1.68	R_T	0.53	0.02
T_b	0.07	1.23	X'_d	0.21	0.45
a_{23}	1.01	0.05	X'_q	0.41	0.13
T'_{do}	0.14	0.94	T_R	0.40	0.02
T_3	0.12	0.77	R_P	0.30	0.01
T_1	0.1	0.77	A_1	0.04	0.03

B. Dataset and Hyper-parameter Setting

To collect the training and test data for the eighteen parameters case, the mean value of the distribution, $\mu_{\pi}(\alpha_c)$, is set as

10% greater than the true values and the lower/upper bounds of the $\pi(\alpha_c)$ are set as 50% less/greater than the $\mu_{\pi}(\alpha_c)$. Multiple parameters are independent, identically distributed random chosen, and with event playback, dynamic generator data generated automatically respective to the critical parameters. Based on the test cases, different simulations number for training the network are used. In this paper, for eighteen parameters, 3×10^4 dynamic simulations are collected to train, validate, and test the model. 90% of the available data is used for training and cross-validation, and 10% is used for testing. For each simulation, the time horizon is set as 20 s. The sampling time is set as 0.033 s, since the reporting rate of PMU measurement is assumed to be 30 samples/s.

The learning rate is set to 10^{-3} . Besides, to avoid over-fitting, L_2 regularization with the regularization coefficient $\lambda = 2$ is used for the dense layers. All the LSTM layers have 100 neurons with the *tanh* and *linear* activation function. The values of the η and ζ are equal to 2. The proposed model is implemented in Python 3.8 and all case studies using the deep learning technique are carried out using a high-level neural networks API, Keras [21]. In this paper, a supervised learning algorithm is used to find the time lags' optimal values t_p and t_q which will be explained in the following section.

C. Higher-Dimensional Case

This case study demonstrates the performance of the trained model in a higher-dimensional case with eighteen critical parameters. The prior distributions for these parameters for training the model are assumed to have a $\mu_{\pi}(\alpha_c)$ with 10% deviation from the true values and follow a uniform distribution as $\mathcal{U}(\mu_{\pi}(\alpha_c) - \mu_{\pi}(\alpha_c) \cdot 50\%, \mu_{\pi}(\alpha_c) + \mu_{\pi}(\alpha_c) \cdot 50\%)$. Table II lists the true and estimated values and the estimation errors verifies the well-trained model has acceptable accuracy for high dimensional parameters.

TABLE II
CALIBRATION OF EIGHTEEN KEY PARAMETERS

Param.	True value	Estimated (% Error)	Param.	True value	Estimated (% Error)
T_6	10	9.99 (0.1)	X_d	0.57	0.57 (0)
K_s	20	20.09 (0.4)	T_c	0.90	0.89 (0.6)
T_5	10	10.03 (0.3)	H	5.4	5.41 (0.1)
K_A	125	124.83 (0.1)	R_T	0.42	0.42 (0)
T_b	3.86	3.86 (0)	X'_d	0.25	0.25 (0)
a_{23}	1.102	1.101 (0.2)	X'_q	0.32	0.32 (0)
T'_{do}	5.4	5.41 (0.1)	T_R	1	1 (0)
T_3	0.15	0.15 (0)	R_P	0.01	0.01 (0)
T_1	0.15	0.154 (2.6)	A_1	0.035	0.0345 (1.4)

In real application, true model parameters are never known. It is possible that for different events, the algorithm may bring different parameter packs. In addition to engineering judgment

and experience that can help choose reasonable parameters [12], techniques can also be developed based on the available multiple events to help select the best parameters. For example, one can consider one event to estimate the parameters and use the other events to cross-validate the estimated parameters. In this paper, it assumes that measurements for four events are available and trained the model with two different events. The remaining two events, were used to estimate the system parameters. The parameters are estimated with each one of the two events and calculate the MSE metric for the other two events. The parameter set with the smallest MSE is selected as the estimated parameters. Fig. 3 shows the model validation results under two events with the original parameters and the estimated best parameters and $\mu_{\pi}(\alpha_c)$. It is seen that the mismatch between model outputs and PMU measurements is significant with $\mu_{\pi}(\alpha_c)$, while the model outputs with the selected estimated parameters can match the PMU measurements very well for the two events.

Fig. 3. Model outputs before and after parameter calibration: (a) Event 1; (b) Event 2. — PMU measurements; - - model outputs before calibration; . . model outputs after calibration.

VI. CONCLUSIONS AND FUTURE WORKS

This paper presents a method to validate stability models, identify critical model parameters, and calibrate them using PMU measurements. With a sensitivity approach, critical parameters of the generator are identified. A PM-LSTM model is trained based on the event playback using different dynamic simulations. The robustness of the proposed model is tested on a hydro generator with eighteen critical parameter case. Simulation results indicate that the proposed method can accurately and efficiently estimate the full distributions of the parameters. Future works will include:

- Developing an improved approach to assess parameter-output behavior from the perspective of nonlinear systems.
- Applying the proposed method to more complex power systems models (like aggregate renewable energy plants and composite load models) with different events.

- Making the proposed method more computationally efficient.

REFERENCES

- [1] S. R. Khazeynasab, J. Zhao, I. Batarseh, and B. Tan, "Power plant model parameter calibration using conditional variational autoencoder," *IEEE Transactions on Power Systems*, 2021.
- [2] S. R. Khazeynasab and J. Qi, "Pmu measurement based generator parameter calibration by black-box optimization with a stochastic radial basis function surrogate model," *NAPS*, 2020.
- [3] A. Boboń, A. Nocoń, S. Paszek, and P. Pruski, "Determination of synchronous generator nonlinear model parameters based on power rejection tests using a gradient optimization algorithm," *Bull. Pol.*, vol. 65, no. 4, 2017.
- [4] L. Fan and Y. Wehbe, "Extended kalman filtering based real-time dynamic state and parameter estimation using PMU data," *Electric Power Systems Research*, vol. 103, pp. 168–177, Oct. 2013.
- [5] R. Huang, R. Diao, Y. Li, J. Sanchez-Gasca, Z. Huang, B. Thomas, P. Etingov, S. Kincic, S. Wang, R. Fan *et al.*, "Calibrating parameters of power system stability models using advanced ensemble kalman filter," *IEEE Trans. Power Syst.*, vol. 33, no. 3, pp. 2895–2905, Oct. 2018.
- [6] Y. Xu, C. Huang, X. Chen, L. Mili, C. H. Tong, M. Korkali, and L. Min, "Response-surface-based Bayesian inference for power system dynamic parameter estimation," *IEEE Trans. Smart Grid*, Jan. 2019.
- [7] Y. Xu, L. Mili, X. Chen, M. Korkali, and L. Min, "A Bayesian approach to real-time dynamic parameter estimation using PMU measurement," *IEEE Trans. Power Syst.*, Sep. 2019.
- [8] S. R. Khazeynasab and J. Qi, "Generator parameter calibration by adaptive approximate bayesian computation with sequential monte carlo sampler," *IEEE Transactions on Smart Grid*, vol. 12, no. 5, pp. 4327–4338, 2021.
- [9] H. Pourmeidani, S. Sheikhfaal, R. Zand, and R. F. DeMara, "Probabilistic interpolation recoder for energy-error-product efficient dbns with p-bit devices," *IEEE Transactions on Emerging Topics in Computing*, 2020.
- [10] H. Pourmeidani, P. Debashis, Z. Chen, and R. F. DeMara, "Process variation sensitivity of spin orbit torque perpendicular nanomagnets in dbns," *IEEE Trans. Magn.*, 2021.
- [11] R. Huang, R. Fan, T. Yin, S. Wang, and Z. Tan, "Parameters calibration for power grid stability models using deep learning methods," *arXiv preprint arXiv:1905.03172*, May 2019.
- [12] S. R. Khazeynasab, J. Qi, and I. Batarseh, "Generator parameter estimation by q-learning based on pmu measurements," *IEEE ISGT NA*, 2020.
- [13] S. Wang, R. Diao, T. Lan, Z. Wang, D. Shi, H. Li, and X. Lu, "A drl-aided multi-layer stability model calibration platform considering multiple events," in *2020 (PESGM)*. IEEE, 2020, pp. 1–5.
- [14] S. Wang, R. Diao, C. Xu, D. Shi, and Z. Wang, "On multi-event co-calibration of dynamic model parameters using soft actor-critic," *IEEE Trans. Power Syst.*, Oct. 2020.
- [15] Z. Huang, P. Du, D. Kosterev, and S. Yang, "Generator dynamic model validation and parameter calibration using phasor measurements at the point of connection," *IEEE Trans. Power Syst.*, vol. 28, no. 2, pp. 1939–1949, Mar. 2013.
- [16] M. Cui, M. Khodayar, C. Chen, X. Wang, Y. Zhang, and M. E. Khodayar, "Deep learning-based time-varying parameter identification for system-wide load modeling," *IEEE Trans. Smart Grid*, vol. 10, no. 6, pp. 6102–6114, Jan. 2019.
- [17] M. Khodayar and J. Wang, "Probabilistic time-varying parameter identification for load modeling: A deep generative approach," *IEEE Trans. Ind. Informat.*, vol. 17, no. 3, pp. 1625–1636, Feb. 2020.
- [18] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
- [19] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, "Short-term residential load forecasting based on lstm recurrent neural network," *IEEE Trans. Smart Grid*, vol. 10, no. 1, pp. 841–851, Sep. 2017.
- [20] M. Dabbaghjamanesh, A. Moeni, N. D. Hatziaargyriou, and J. Zhang, "Deep learning-based real-time switching of hybrid ac/dc transmission networks," *IEEE Trans. Smart Grid*, Dec. 2020.
- [21] F. Chollet *et al.*, "keras," 2015. [Online]. Available: <https://github.com/fchollet/keras>