U.P.B. Sci. Bull., Series C, Vol. 88, Iss. 3, 2026 ISSN 2286 – 3540

CONFIDENCE ANALYSIS AND DEEP LEARNING-BASED APPROACH FOR TWO-STAGE CALIBRATION OF ERROR-DOMINANT PARAMETERS IN POWER SYSTEM SIMULATION

Shengliang CAI[1], Guobin FU[1], Mingzhi CHEN [2], Guoqiang LU[3], Haibin SUN[1], Xuebin WANG[1], Rui SONG[1], Yanzhe CHENG[2, *]

Accurate identification and calibration of error-dominant parameters (EDPs) are critical for reliable power-system simulations. This paper presents a deep-learning-based framework for unified EDP identification and calibration. A large synthetic dataset is constructed by perturbing model parameters and recording the corresponding trajectory responses. A trajectory-similarity metric is employed to train a multilayer perceptron that maps similarity patterns to parameter perturbations, while confidence analysis is used to identify physically plausible EDPs. A dual-branch CNN-Transformer network then extracts time- and frequency-domain features, learning a direct mapping to calibrated EDPs. Tests on the IEEE New England 39-bus system show that the framework better captures the joint influence of multiple parameters, improves EDP selection completeness, and addresses local-minima issues in reinforcement-learning-based calibration.

Keywords: Power System Dynamic Simulation; Trajectory Similarity; Parameter Identification and Calibration; Deep Learning; Confidence Analysis

1. Introduction

Numerical simulation is a fundamental tool for power system planning, operation, and security assessment. However, inaccuracies in model parameters can lead to significant discrepancies between simulated trajectories and actual system responses, as evidenced by large-scale blackouts such as the Western North America event and subsequent disturbances [1–3]. These discrepancies undermine the credibility of simulation-based analyses and motivate systematic parameter calibration to improve the fidelity of power system dynamic models [4].

A key challenge is to select error-dominant parameters (EDPs) from a large candidate set while controlling computational cost. Most existing studies address this by analyzing the sensitivity trajectories of individual parameters with respect to error variables. For example, [5] evaluates sensitivity trajectories and uses first-swing amplitude and oscillation damping as intuitive indicators for rapid EDP

[1] Electric Power Research Institute, State Grid Qinghai Electric Power Company, Qinghai 810001, China
[2] School of Electrical Engineering, Northeast Electric Power University, Jilin 132012, China
[3] Dispatch and Control Center, State Grid Qinghai Electric Power Company, Qinghai 810001, China
* Correspondent author, e-mail: 13263512305@163.com

selection, while [6] assigns different weights to trajectory features over different time intervals. Other works [7,8] transform sensitivity trajectories into the frequency domain to describe how each parameter affects error variables from a spectral perspective. However, sensitivity-based EDP identification often fails to accurately reflect parameter importance, as parameters are evaluated independently and the analysis relies on hand-crafted indices. This can result in inaccurate EDP selection and reduced calibration effectiveness.

Intelligent optimization algorithms, such as Particle Swarm Optimization (PSO), are commonly used for EDP calibration due to their simplicity and computational efficiency [9-11]. These methods minimize the error between simulated and measured values, but their performance heavily depends on parameter bounds, constraints, and penalty terms. Additionally, calibration based on a small set of variables at fault locations may not fully represent system dynamics and is sensitive to measurement noise, leading to ill-posed solutions. To address this, methods like Kalman filtering and reinforcement learning (e.g., SAC, DQN, DDPG) have been introduced but still face challenges such as high computational cost and unstable results [12-17].

Recent advances in deep learning have demonstrated strong capability in extracting representative features from large-scale, high-dimensional data, leading to increasing applications in power system analysis [18–21]. By leveraging multi-bus measurements, deep learning models can capture complex spatiotemporal dependencies and improve robustness under noisy conditions. However, most existing studies focus on stability assessment or state estimation, while deep-learning-based frameworks that explicitly integrate EDP identification with parameter calibration remain limited.

This paper proposes a deep-learning-based framework for EDP calibration that integrates time- and frequency-domain feature extraction. A multilayer perceptron (MLP) first learns the mapping between parameter perturbations and their impact on trajectories. The network's outputs are used to quantify EDP confidence and assess calibration needs. Once the EDP set is determined, Convolutional Neural Networks (CNNs) and Transformers are employed to model the relationship between variable trajectories and EDP values, thereby enabling data-driven parameter calibration.

The main contributions of this study, in comparison with existing methods, are summarized as follows:

1) Deep-learning-based EDP identification using trajectory similarity. A trajectory-similarity index quantifies relationships among trajectories, and a neural network maps similarity patterns to parameter perturbation rates, thereby improving EDP identification accuracy.
2) Uncertainty-aware EDP selection without manual thresholds. Probabilistic weighting constructs an efficient training set. MC Dropout obtains predictive

distributions, and uncertainty analysis identifies high-confidence EDPs, reducing computational cost and avoiding subjective threshold selection.

3) Reformulating EDP calibration as a supervised regression problem. Instead of relying on iterative optimization, the proposed method performs supervised learning on large collections of trajectories annotated with parameter values, explicitly treating EDP calibration as a regression task. This formulation improves utilization of available measurements and yields calibration results that better reflect global system behavior.
4) A dual-branch CNN–Transformer architecture for trajectory feature fusion. The network captures frequency-domain and temporal features of trajectories, and their fusion enhances the stability and reliability of calibration results.

The remainder of this paper is organized as follows. Sections 2 and 3 describe the identification and calibration of EDPs, respectively. Section 4 reports the case study on the IEEE New England 39-bus system. Section 5 concludes the paper and outlines directions for future work.

2. Intelligent Identification of EDPs

In power systems, the tight interconnection among components leads to strong coupling among electrical variables. Small variations in parameters can induce markedly different responses in multi-bus variable trajectories. After a disturbance, the system dynamics can be described by a set of differential–algebraic equations (DAEs), as shown in Equation (1). Here, t represents time, $x(t)$ denotes the state variables in the system, $y(t)$ represents the algebraic variables, and a indicates the parameters in the system. Variations in the system parameters a induce corresponding responses in the variables $x(t)$ and $y(t)$.

$$\begin{cases} \dot{x}(t) = f(x(t), y(t), a) \\ 0 = g(x(t), y(t), a) \end{cases} \tag{1}$$

Currently, wide-area measurement systems (WAMS) provide high-resolution PMU data that capture rapid system dynamics and support stability monitoring. The goal of EDP identification is to filter D_{par} to reduce the dimensionality of parameter calibration. This section presents the EDP identification procedure, with emphasis on sampling data processing, training set construction, and the proposed confidence-based identification method.

2.1. Definition of Variable Trajectory Similarity

Deep-learning-based EDP identification requires extensive parameter perturbations, leading to large-scale multi-bus time-series data whose direct feature extraction incurs substantial computational burden. To obtain a compact representation of trajectory discrepancies, the Theil Inequality Coefficient (TIC) is adopted.

$$U = \frac{\sqrt{\frac{1}{n}\sum_{t=1}^{n}(Z_t - \hat{Z}_t)^2}}{\sqrt{\frac{1}{n}\sum_{t=1}^{n}(Z_t)^2} + \sqrt{\frac{1}{n}\sum_{t=1}^{n}(\hat{Z}_t)^2}} \tag{2}$$

In Equation (2), the time-series trajectory of each variable obtained from pre-calibration simulations is defined as the reference trajectory Z_t, whereas $\hat{Z}_t$ denotes the corresponding trajectory after parameter perturbation, and n denotes the length of the time series. Considering the damping characteristics of post-disturbance responses, discrepancies are typically more pronounced during the initial oscillation stage. Accordingly, a Gaussian weighting function is introduced in Equation (2) to emphasize early-stage differences, where d represents the oscillation-duration-based scaling factor and k is set to 1. The final trajectory similarity formula adopted after modification is:

$$\hat{U} = \frac{\sqrt{\frac{1}{n}\sum_{t=1}^{n}(de^{-t^2}Z_t - de^{-t^2}\hat{Z}_t)^2}}{\sqrt{\frac{1}{n}\sum_{t=1}^{n}(Z_t)^2} + \sqrt{\frac{1}{n}\sum_{t=1}^{n}(\hat{Z}_t)^2}} \tag{3}$$

$$A = \begin{bmatrix} \hat{U}_{1\theta_1} \cdots \hat{U}_{1\theta_i} & \hat{U}_{1V_1} \cdots \hat{U}_{1V_i} & \hat{U}_{1P_1} \cdots \hat{U}_{1P_i} & \hat{U}_{1Q_1} \cdots \hat{U}_{1Q_i} & \hat{U}_{1f_1} \cdots \hat{U}_{1f_i} \\ \hat{U}_{2\theta_1} \cdots \hat{U}_{2\theta_i} & \hat{U}_{2V_1} \cdots \hat{U}_{2V_i} & \hat{U}_{2P_1} \cdots \hat{U}_{2P_i} & \hat{U}_{2Q_1} \cdots \hat{U}_{2Q_i} & \hat{U}_{2f_1} \cdots \hat{U}_{2f_i} \\ \cdots & & & & \\ \hat{U}_{m\theta_1} \cdots \hat{U}_{m\theta_i} & \hat{U}_{mV_1} \cdots \hat{U}_{mV_i} & \hat{U}_{mP_1} \cdots \hat{U}_{mP_i} & \hat{U}_{mQ_1} \cdots \hat{U}_{mQ_i} & \hat{U}_{mf_1} \cdots \hat{U}_{mf_i} \end{bmatrix} \tag{4}$$

$$B = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1z} \\ r_{21} & r_{22} & \cdots & r_{2z} \\ \cdots & & & \\ r_{m1} & r_{m2} & \cdots & r_{mz} \end{bmatrix} \tag{5}$$

Using the defined trajectory similarity $\hat{U}$, the original dataset of size $x \times y \times m$ is reduced to a dataset of size $x \times m$, forming a matrix A with x rows and m columns. Here, the number of rows represents the number of perturbations, while the number of columns corresponds to the total number of measured bus variables. The matrix structure is shown in Equation 4. For matrix A, the parameter variation vector r associated with each perturbation is organized into matrix B, where the dimension z depends on the number of elements in D_{par}.

2.2. Data Input Form for EDPs Identification Module

Trajectory similarity provides a compact descriptor that enables efficient learning from large-scale multi-bus trajectories. However, it is inherently a compressed representation of the original spatiotemporal response. As a result, the inverse mapping from similarity to parameter deviations can be ill-posed: different trajectory sets may yield similar (or even identical) similarity values, while corresponding time-series distributions still differ, which leads to non-unique and dispersed prediction outcomes. In this work, we therefore emphasize that the Stage-1 mapping model is not intended to serve as a deterministic estimator for accurate parameter deviation recovery. Instead, it is designed as a confidence-aware screening module to identify EDPs under a given similarity feature space.

Specifically, we exploit the fact that the training data are densest around small perturbations and adopt a tolerance-based loss to ensure reliable behavior near $r = 0$. Then, MC Dropout is used to obtain predictive distributions rather than single-point estimates. Parameters whose output distributions exhibit persistent deviation from zero and abnormally wide confidence intervals are interpreted as being insufficiently explained by similarity features and are therefore classified as EDPs. Once the EDP set is determined, Stage-2 performs supervised calibration using richer time-domain and frequency-domain trajectory features, which alleviates the ambiguity introduced by similarity compression.

Parameter perturbation requirements are defined as follows. The perturbation value p_m ranges from 60% to 140% of the initial value p_0. Three sets of probability density functions (PDFs) are defined along with their applicable intervals:

$$f(r) = \begin{cases} C_1 & 0.6 \le r\% \le (1-k_s) \\ C_2 \cdot Beta(\dfrac{r\% - 1 + k_s}{2k_s};5,5) & (1-k_s) \le r\% \le (1+k_s) \\ C_3 & (1+k_s) \le r\% \le 1.4 \end{cases} \tag{6}$$

In the equation, C_1, C_2, and C_3 are the probability density constants for the three intervals. The term $C_2 \cdot Beta$ defines the PDF in the vicinity of $r = 0$, following a Beta distribution. The interval $(1-k_s\%, 1+k_s\%)$ defines the range around $r = 0$, and the Beta distribution has shape parameters of $Beta(5,5)$. For the intervals farther from $r = 0$, the probability densities are C_1 and C_3, which follow a uniform distribution. Typically, C_1=C_3 . To ensure the normalization of the entire probability distribution, the cumulative probabilities of each interval must satisfy:

$$\int_{0.6}^{1.4} f(r)dr = 1 \tag{7}$$

2.3. Confidence Analysis of EDPs Based on MLP

Matrices A and B are constructed based on the quantitative evaluation of variable trajectory similarity and the standardized parameter perturbation data. By learning the mapping relationship between these two matrices, parameters in D_{par}

can be further screened from a system-wide perspective. The architecture of the neural network is shown in Fig. 1. It consists of a one-dimensional (1D) input layer, three Dense layers, a user-defined module (UDM), and an output layer corresponding to EDPs.

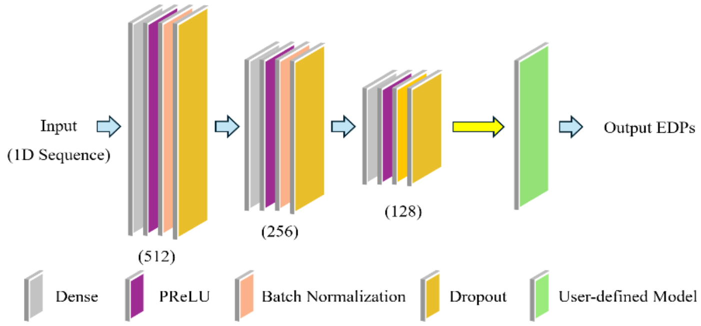

Fig. 1. Trajectory similarity to parameter perturbations mapping model

The input is a 1D data set that includes negative values. To ensure the gradient flow for negative values, the PreLU function is used as the activation function for the Dense layers. This activation function enables adaptive responses in the negative input region based on data characteristics, thereby enhancing the network's capacity to learn and represent complex features.

Batch Normalization standardizes the activation values of intermediate layers, improving training efficiency and model performance. The process is implemented by calculating the mean μ_{B} and variance σ_{B}^2 of the activation values for each dimension within a mini batch. These activation values are then normalized, and learnable scaling parameter γ_i and shifting parameter β_i are introduced for adjustment.

The purpose of this module is to identify EDPs rather than to estimate precise parameter perturbation rates. Accordingly, a tolerance-based loss function is defined as follows:

$$loss_{i,j}^{1} = \max(error_{i,j} - \tau_j, 0) \tag{8}$$

Here, $error_{i,j} = \left| y_{true,i,j} - y_{pred,i,j} \right|$. For each output dimension j and sample i, a loss is incurred only when the error exceeds the tolerance τ_j; otherwise, the loss is 0. Therefore, the final loss is calculated as shown in Equation (9), where N represents the number of samples and M represents the output dimensions.

$$L_{def} = \frac{1}{N \times M} \sum_{i=1}^{N} \sum_{j=1}^{M} loss_{i,j} \tag{9}$$

When the model is trained based on this loss function, the output is an approximation of the parameter deviation rate r within the range $\pm\tau_j$, which significantly reduces computational cost and improves convergence.

The uncertainty of the prediction results is determined by the inherent characteristics of the data. To further assess the model predictions, UDM is added after the Dense layer in the model. The function of this module is to evaluate the confidence of the output results. Specifically, the MC Dropout method is employed, where multiple forward passes are conducted with fixed input data (the similarity data of the measured trajectories). By activating different neurons and connections in each pass, the distribution characteristics of the prediction results are analyzed.

As shown in Equation 10, $\hat{Y}_t$ represents the different prediction results generated by using different dropout patterns $R^{(t)}$ during each forward pass. MC Dropout outputs a set of parameter deviation values r for each iteration, and after k iterations, a data distribution model can be constructed based on the outputs from each pass.

$$\hat{Y}_t = \text{Model}(X, R^{(t)}) \quad t = 1, 2, \dots, m \tag{10}$$

The output values' PDFs can be mainly classified into unimodal and multimodal distributions, with varying ranges of probability densities. This behavior can be attributed to two factors. First, after sufficient training of the trajectory-similarity-based prediction model, certain parameters may admit multiple plausible outputs; second, when certain parameters significantly affect the error variables, their r values tend to deviate from 0. Furthermore, due to the bias in the training data, these parameters' probability density intervals become wider after multiple MC Dropout iterations.

Before analyzing the probability of the parameter output values, an uncertainty analysis is first performed on the trained model using the MC Dropout method to ensure that the model can accurately capture predictions for parameters near $r = 0$. A random selection of n_1 sets of parameters is made from the validation set, with their feature variables fixed. Then, k_1 forward passes are conducted to obtain the predicted range $r_{\min} \sim r_{\max}$ for each parameter when $r < k_s$. The coverage rate ξ of the true value range $r_t \in [r_{\min}, r_{\max}]$ for each target variable is calculated based on the results of these n_1 sets of data. When ξ approaches 1, the parameter feature under the condition $r < k_s$ is considered identifiable, enabling further assessment of feature resolution based on the output distribution of r. If the coverage rate ξ for specific parameters shows significant deviation, it suggests feature interference during model training. Accordingly, a significant deviation in ξ indicates that the corresponding parameter features are not adequately captured by the model. Without further model refinement, such parameters are therefore classified as EDPs.

For the parameters that satisfy the conditions for ξ, kernel density estimation (KDE) is used to analyze the probability density of the prediction results:

$$\hat{f}(x) = \frac{1}{nh\sqrt{2\pi}} \sum_{i=1}^{n} \exp(-\frac{(x - X_i)^2}{2h^2}) \tag{11}$$

Here, X_i represents the sample, and h is the bandwidth. For parameters with high feature resolution, the range of the output interval CI is relatively small. However, for EDPs, the parameter r will show a significant deviation from 0. Due to the influence of the training data distribution, the feature recognition of these parameters will suffer from underfitting, leading to a significantly larger range for the CI values.

3. Intelligent Calibration of EDP

The identification of EDP relies on the evaluation of variable trajectory similarity. However, this is not sufficient to fully capture both the temporal and spatial information of the trajectories. Therefore, further exploration of the spatiotemporal characteristics of the variable trajectories is required.

3.1 Time-Frequency Transformation of Data

The Discrete Fourier Transform (DFT) is commonly used to convert discrete time-series signals into frequency-domain signals, as shown in Equation 12. Here, $\hat{X}_k$ represents the amplitude components in the frequency domain, and k_D is the index of the frequency.

$$\hat{X}_{k_D} = \sum_{t=0}^{n-1} x(t) e^{-j2\pi \frac{k_D t}{n}}, \quad k_D = 0, 1, \dots, N-1 \tag{12}$$

After multiple random perturbations of the EDPs, the DFT results of the multidimensional variable data are obtained. Based on the frequency-domain data and the EDP values, a dataset in the format suitable for CNN training is constructed.

3.2 CNN+Transformer-Based Intelligent EDP Calibration

Unlike traditional optimization methods, deep learning formulates EDP calibration as a supervised regression problem by learning the mapping between trajectory features and parameter values from perturbation-generated data. To enhance feature representation, a dual-branch CNN–Transformer architecture is adopted to separately extract frequency-domain and temporal features and fuse them for improved calibration accuracy and robustness, as illustrated in Fig. 2.

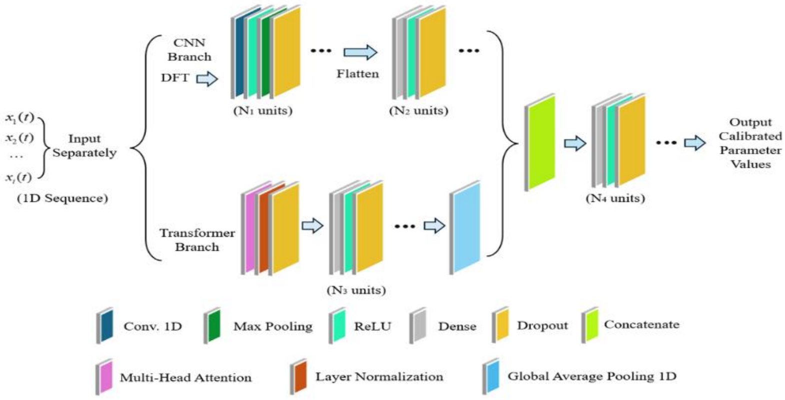

Fig. 2. CNN+Transformer -based EDP Calibration Model

Owing to temporal dependencies in power system operating states, the multi-head self-attention mechanism in the Transformer effectively integrates temporal information X_{time}. By computing Query, Key, and Value, it effectively captures essential information. The attention mechanism computes the scaled dot product of Q and K, applies the function to obtain the attention weights, and then uses these weights to compute the weighted output for V:

$$\text{Attention}_{output} = \text{SoftMax}(\frac{QK^T}{\sqrt{d_k}})V \tag{13}$$

Here, d_k denotes the dimension of each attention head. Although the Transformer is data-driven, its self-attention weights are explicitly computed and can be inspected. Specifically, the SoftMax term in (13) produces normalized relevance scores over time steps, which can be interpreted as an adaptive weighting of post-disturbance intervals. In the context of power-system transients, this provides a transparent mechanism to emphasize physically informative segments while down-weighting less informative quasi-steady parts. Therefore, the Transformer branch does not rely on opaque latent factors only; it also yields an interpretable relevance pattern through attention weights. The outputs of multiple attention heads are concatenated and linearly transformed to produce the final representation. Frequency-domain features extracted by the CNN branch and time-domain features extracted by the Transformer branch are then fused through a feature fusion layer, as defined in Equation (14).

$$F_{merged} = Concat(F_{CNN}, F_{Transformer}) \quad F_{merged} \in \mathrm{R}^{N\times(d_1+d_2)} \tag{14}$$

The two branches are designed to align with complementary and physically meaningful characteristics of post-disturbance trajectories. The CNN branch operates on frequency-domain representations obtained via the DFT in Section 3.1, where convolutional filters learn spectral patterns that correspond to oscillatory signatures. In parallel, the Transformer branch models the temporal evolution of the transient process via attention-based weighting. The fusion in (14) thus combines "how the trajectory evolves in time" with "which oscillatory components dominate in frequency," providing a structured and physically interpretable basis for EDP calibration. Improving the alignment between predicted and actual parameter values requires minimizing prediction errors, since certain deviations can significantly affect system operation. The use of Mean Squared Error (MSE) introduces a stronger penalty for large deviations, prompting the model to focus on accurately predicting critical parameters. Once the model is trained, the measured feature variables are directly input, providing the final EDP calibration results. By leveraging feature information from multi-bus data, the model learns a more comprehensive data distribution and feature representation than traditional calibration methods, which improves generalization and mitigates localized information bias.

4. Test Results

To assess the impact of model uncertainty, the IEEE New England 39-bus system (Fig. 3) is adopted as a case study to validate the reliability of the proposed EDP calibration method.

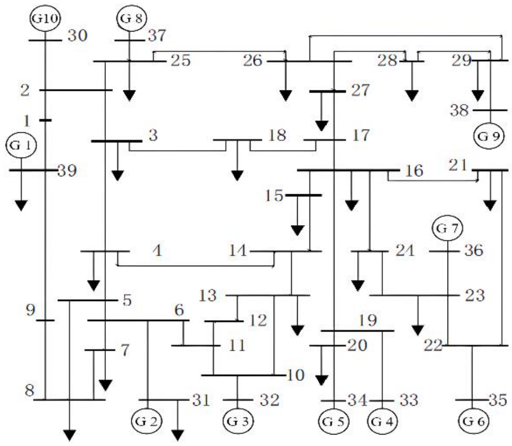

Fig. 3. Connection diagram of the IEEE New England 39-bus system

Standard data of the 39-bus system are used for model initialization, whereas actual system parameters are randomly perturbed within 70–130% of their pre-calibration values to represent long-term operational variations. PSAT is employed for simulation, where synchronous machines use the IV-order model, excitation systems the II-order model, and loads the ZIP model. A three-phase short-circuit fault is applied at Bus 2 at 0 s and cleared at 0.1 s.

In practical applications, WAMS/PMU measurements are inevitably subject to noise and measurement errors. To emulate realistic operating conditions, additive

Gaussian noise is introduced into the simulated trajectories. Specifically, for each trajectory $Z(t)$, the noisy measurement is defined as

$$Z^m(t) = Z(t) + N(0, \sigma_Z^2) \tag{15}$$

Computational results indicate that the average R^2 for $\Delta\theta$ is $R_{avg}^2 = 0.849$, $R_{\min}^2 = 0.808$ ($\Delta\theta_{\text{Bus\_34}}$) ; U is $R_{avg}^2 = 0.992$, $R_{\min}^2 = 0.969$ ($U_{\text{Bus\_34}}$) ; P is $R_{avg}^2 = 0.866$, $R_{\min}^2 = 0.682$ (P_{SM_5}); Q is $R_{avg}^2 = 0.978$, $R_{\min}^2 = 0.905$ (Q_{SM_5}); and ω is $R_{avg}^2 = 0.834$, $R_{\min}^2 = 0.712$ (ω_{SM_5}). According to the computational results, the R_{avg}^2 values for $\Delta\theta$, P, and ω exhibit significant deviations. Furthermore, while the R_{avg}^2 for Q remains within acceptable limits, its $R_{\min}^2$ is substantially below the average value. These observations confirm that $\Delta\theta$, P, Q, and ω exhibit notable discrepancies between simulated and measured values.

Before the calculations, a residual-based approach is employed to quantify discrepancies between measured and simulated nodal variables and to identify specific error variables. These include: 38 sets of time-series phase angle $\Delta\theta = \theta_{Bus\_i} - \theta_{Bus\_1}$ data, 39 sets of time-series bus voltage U data, and 10 sets each of time-series active power P, reactive power Q, and rotor angular velocity ω data from 10 synchronous machines. The union of error-sensitive parameters derived from variables $\Delta\theta$, P, Q, and ω yields 28 elements. The D_{par} set includes { $EXC_1\_K_w$; SM_1_M; SM_2_M; SM_3_M; SM_4_M; SM_5_M; SM_6_M; SM_7_M; SM_8_M; SM_9_M; SM_{10}_M; $SM_1\_T'_{d0}$; $SM_5\_X_d$; $SM_9\_X_d$; $SM_1\_X'_d$; $SM_3\_X'_d$; $SM_5\_X'_d$; $SM_6\_X'_d$; $SM_8\_X'_d$; $SM_9\_X'_d$; $SM_{10}\_X'_d$; $SM_1\_X_q$; $SM_8\_X_q$; $SM_9\_X_q$; $SM_1\_X'_q$; $SM_8\_X'_q$; $SM_9\_X'_q$; $SM_{10}\_X'_q$ }. The parameter definitions adhere to standardized power system modeling conventions: SM denotes synchronous machine parameters (with numerical subscripts indicating generator numbering, e.g., SM_1 for Generator 1), EXC represents excitation system parameters, M is the inertia time constant, X_d and X_q correspond to d-axis and q-axis synchronous reactance, X'_d and X'_q indicate d-axis and q-axis transient reactance, T'_{d0} specifies the d-axis sub transient open-circuit time constant, and K_w denotes the washout gain coefficient in the excitation control loop.

4.1 EDP Identification Test

To identify calibration-relevant parameters in D_{par}, multivariable trajectory features are analyzed after defining $k_s = 5$ and applying extensive random perturbations. The resulting trajectory similarity is used as the input feature, and the parameter variation rate as the target, to train a pre-constructed MLP with three Dense layers (512, 256, and 128 neurons). The tolerance is set to $\tau_j = 1$, such that the loss is zero when the absolute prediction error is ≤ 1.

Table 1.

Coverage probability ξ of true parameter values

Parameter ξ (%)	Parameter ξ (%)	Parameter ξ (%)	Parameter ξ (%)
$EXC_1\_K_w$ 100.0	SM_7_M 99.0	$SM_1\_X'_d$ 94.2	$SM_1\_X_q$ 99.2
SM_1_M 100.0	SM_8_M 98.4	$SM_3\_X'_d$ 95.8	$SM_8\_X_q$ 96.0
SM_2_M 100.0	SM_9_M 100.0	$SM_5\_X'_d$ 96.0	$SM_9\_X_q$ 96.0
SM_3_M 98.8	SM_{10}_M 98.4	$SM_6\_X'_d$ 94.8	$SM_1\_X'_q$ 98.6
SM_4_M 97.4	$SM_1\_T'_{d0}$ 96.8	$SM_8\_X'_d$ 94.6	$SM_8\_X'_q$ 98.4
SM_5_M 99.4	$SM_5\_X_d$ 100.0	$SM_9\_X'_d$ 95.2	$SM_9\_X'_q$ 95.0
SM_6_M 99.0	$SM_9\_X_d$ 100.0	$SM_{10}\_X'_d$ 97.0	$SM_{10}\_X'_q$ 98.4

After model training, uncertainty analysis is conducted using MC dropout. 500 data samples are randomly selected from the validation set, and the ξ values of parameters satisfying $r < k_s$ are calculated for each sample. Each data sample undergoes 1000 Monte Carlo dropout iterations. In terms of computational cost, the above configuration corresponds to 5×10^5 stochastic forward passes of the lightweight MLP. On a workstation with an Intel Core i9 (14th Gen) CPU and an NVIDIA RTX 4090 GPU, the above MC Dropout evaluation takes approximately 1 minute, with peak GPU memory consumption under 1 GB. In addition, this MC Dropout procedure is conducted offline for confidence analysis and EDP screening, whereas the subsequent CNN–Transformer calibration is a standard single-pass inference at deployment time. The computational results are shown in Table 1. In this case study, all parameters achieve coverage probabilities ξ of at least 94.2%, indicating acceptable prediction accuracy as r approaches 0. Parameters with well-covered r values are then further analyzed through probability density estimation.

Fig. 4 illustrates the probability density histograms of all 28 parameters, where shaded regions denote the 95% confidence intervals $CI_{95\%}$, and “Mean” and “STD” represent the output mean and standard deviation, respectively.

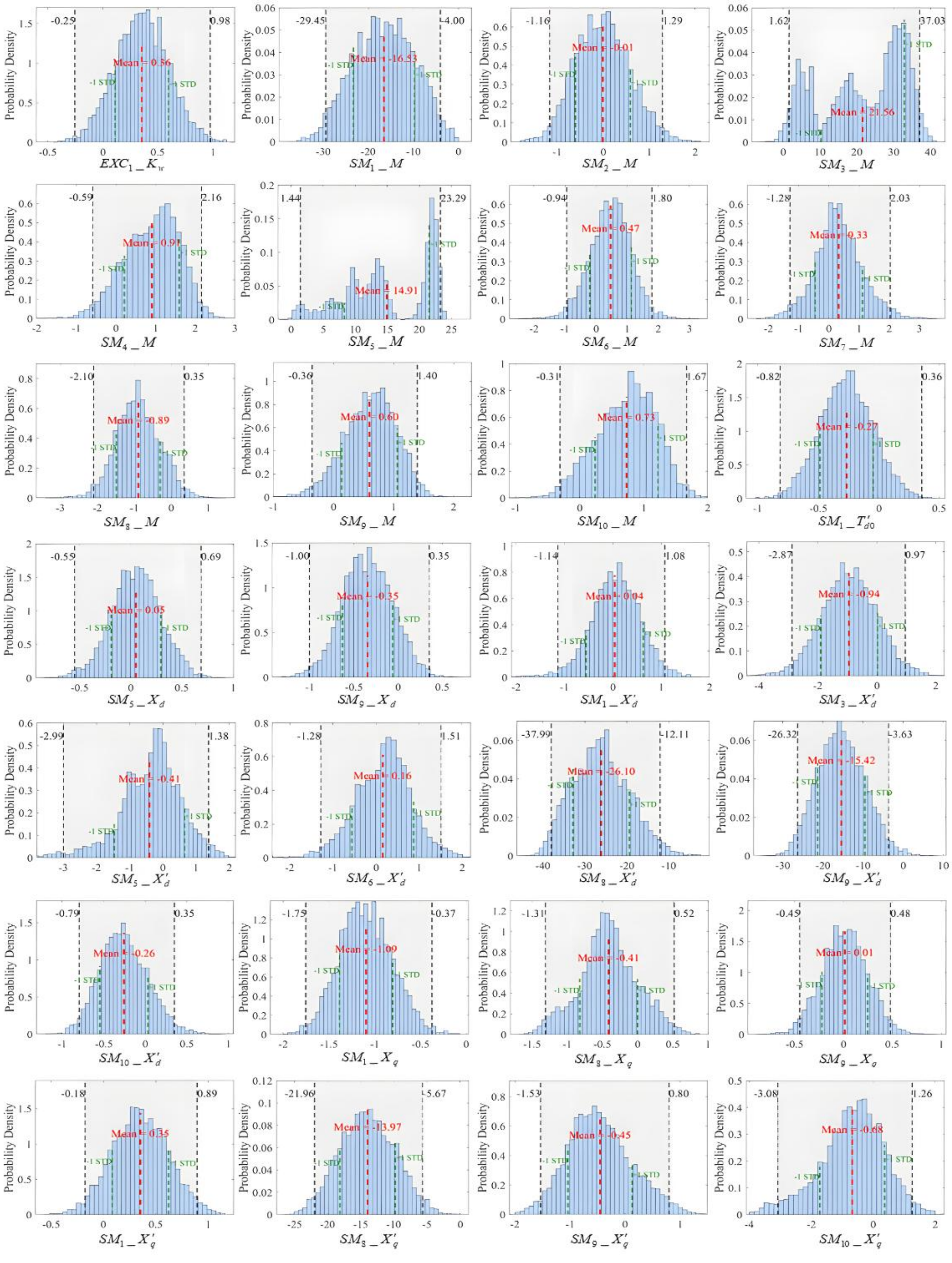

Fig. 4. Probability density distributions of r outputs

The mean values of SM_1_M, SM_3_M, SM_5_M, $SM_8\_X'_d$, $SM_9\_X'_d$ and $SM_8\_X'_q$ deviate significantly from zero, and their confidence intervals differ markedly from those of other parameters. This is attributed to their corresponding rrr values being far from zero, leading to underfitting and increased uncertainty.

These parameters are therefore classified as EDPs. The final screened EDPs are SM_1_M, SM_3_M, SM_5_M, $SM_8\_X'_d$, $SM_9\_X'_d$, and $SM_8\_X'_q$, eliminating 78.6% of redundant parameters compared with the sensitivity-based approach.

4.2 Trajectory Calibration

The calibration of EDPs is carried out using a dual-branch neural network that jointly exploits time- and frequency-domain features. The CNN branch comprises three convolutional layers with 32, 64, and 128 filters, respectively, and uses max-pooling (window size = 2) to extract hierarchical frequency-domain representations. The Transformer branch captures temporal dependencies and dynamic characteristics in time-domain signals by employing eight attention heads (key dimension = 64) and projecting the inputs into a 128-dimensional space via a fully connected layer. The features extracted by the two branches are then fused in a feature-fusion layer and passed through fully connected layers to produce the final EDP calibration results. Perturbations were applied to the 6 selected parameters (SM_1_M , SM_3_M , SM_5_M , $SM_8\_X'_d$, $SM_9\_X'_d$, and $SM_8\_X'_q$) within a range of 60% to 140% of their original values, generating a valid dataset of 19,963 samples. The deep neural network framework employed in this study possesses adaptive feature fusion capabilities. By establishing multidimensional trajectory-to-parameter mapping relationships in a nonlinear space, the proposed framework mitigates the limitations of conventional calibration methods that rely on single-trajectory data or manual weight allocation. In this case study, all measurement data from buses near the fault location are utilized as input, including P, Q, ω, and $\Delta\theta$ from Buses 2, 3, 25, and 30.

As shown in Table 2, the proposed CNN-Transformer model produces parameter estimates that are consistently closer to the measured values than those obtained by PSO, a single-branch Transformer, or a single-branch CNN. For the inertia parameters SM_1_M and SM_3_M, the CNN-Transformer achieves relative errors below 2%, whereas PSO, Transformer, and CNN exhibit noticeably larger deviations. For the reactance-related parameters $SM_8\_X'_d$, $SM_9\_X'_d$, and $SM_8\_X'_q$, the CNN-Transformer yields the smallest absolute error in four out of six cases and avoids the severe overestimation observed for PSO and the single-branch models, where the relative error can approach or exceed several tens of percent.

Table 2.

Parameter calibration results across methodologies

Parameter	CNN+ Transformer	PSO	Transformer	CNN	Measure Value
SM_1_M	762.0486 s	735.6500 s	898.7760 s	790.3708 s	756.2394 s
SM_3_M	88.1813 s	98.8400 s	102.1021 s	72.7710 s	89.8453 s

SM_5_M	48.3648 s	45.2080 s	69.7644 s	45.4341 s	43.2278 s
$SM_8\_X'_d$	0.0431 p.u.	0.0798 p.u.	0.0712 p.u.	0.0461 p.u.	0.0405 p.u.
$SM_9\_X'_d$	0.0496 p.u.	0.0580 p.u.	0.0408 p.u.	0.0454 p.u.	0.0445 p.u.
$SM_8\_X'_q$	0.0502 p.u.	0.0342 p.u.	0.0630 p.u.	0.0743 p.u.	0.0473 p.u.

Overall, the CNN-Transformer attains the lowest average relative error (about 6%) among all methods and provides the most balanced performance across both inertia and reactance parameters, highlighting the benefit of jointly exploiting time- and frequency-domain features for EDP calibration.

5. Conclusions

This paper presents a deep-learning-based framework for identifying and calibrating error-dominant parameters (EDPs) in power system dynamic simulations. Using trajectory similarity and uncertainty analysis, the identification stage removes redundant parameters while retaining those that significantly affect simulation error, thereby reducing the search space without compromising physical relevance. Based on the reduced EDP set, a dual-branch CNN–Transformer network integrating time- and frequency-domain features from multi-bus trajectories achieves lower calibration errors and more stable performance than PSO and single-network regression models on the IEEE New England 39-bus system. The results show that data-driven, confidence-aware EDP selection combined with supervised regression enhances the credibility of dynamic simulations and offers a practical alternative to traditional sensitivity-based and heuristic optimization methods.

Acknowledgment

This research was funded by the Science and Technology Project of State Grid Qinghai Electric Power Company. The project, titled "Research on Parameter Verification Methods for Numerical Simulation Model Parameters of Grid-Connected Power Systems with High Penetration of Renewable Energy," was supported under grant number 522800240006.

R E F E R E N C E S

[1]. *I. A. Hiskens*, "Nonlinear Dynamic Model Evaluation from Disturbance Measurements", IEEE Power Engineering Review, vol. 21, no. 10, 2007, p. 60.

[2]. *I. A. Hiskens* and *M. Akke*, "Analysis of the Nordel power grid disturbance of January 1, 1997 using trajectory sensitivities", IEEE Transactions on Power Systems, vol. 14, no. 3, 1999, pp. 987-994.

[3]. *C. Tsai, L. Chang-Chien, I.-J. Chen, J. Lin, W. Lee, C. Wu et al.*, "Practical Considerations to Calibrate Generator Model Parameters Using Phasor Measurements", IEEE Transactions on Smart Grid, vol. 8, no. 5, 2017, pp. 2228-2238.

[4]. *L. L. Nascimento, M. Moreto,* "A Systemic Approach for Identifying Parameter Errors in Simulation Models of Power Systems Using Machine Learning", Journal of Control, Automation and Electrical Systems, vol. 36, no.8, 2025, pp. 970-980.

[5]. *D. Zonetti, R. Ortega, R. Cisneros, et al.,* "An observer-based composite identifier for online estimation of the Thévenin equivalent parameters of a power system", Asian Journal of Control, vol. 5, no.3, 2024, pp. 2229-2241.
[6]. *S. R. Khazeiynasab, J. Zhao* and *I. Batarseh.,* "Power Plant Model Parameter Calibration Using Conditional Variational Autoencoder", IEEE Transactions on Power Systems, vol. 37, no.2, 2022, pp. 1642-1652.
[7]. *K. Jae-Kyeong, H. Kyeon,* "On the perturbation size of the finite difference method for trajectory sensitivity-based assessment of power system dynamics with non-smooth behavior", International Journal of Electrical Power and Energy Systems, vol. 155, no. 11, 2023, pp. 109647.
[8]. *M. Bahramgiri, M. Ehsan, B. S. Mozafari,* "Multi-objective dynamic VAR planning against fault-induced delayed voltage recovery using heuristic optimization", Electrical Engineering, vol. 106, no. 5, 2024, pp. 6281-6293.
[9]. *R. A. Pinedo, F. C. Esquivel, P. A. Martínez, et al.,* "Selective optimal under-frequency load shedding based on trajectory sensitivities", Electric Power Systems Research, vol. 234, no. 7, 2024, pp. 110810-110810.
[10]. *H. Benbouhenni, N. Bizon,* "Assessing the Effectiveness of an Intelligent Algorithms-Based PII2 Controller in Enhancing the Quality of Power Output from a DFIG-Based Power System", Energies, vol. 18, no. 21, 2025, pp. 5566-5566.
[11]. *H. Zhen, L. Zhao, B. Zhou, A. Zhao, X. Xiang* and *R. Diao*, "Intelligent parameter identification method of DFIG system based on particle swarm optimization and reinforcement learning", Proceedings of the CSU-EPSA, prepublish, 2025, pp. 1-9.
[12]. *M. Matar, B. Xu* and *R. Elmoudi.,* "A Deep Learning-Based Framework for Parameters Calibration of Power Plant Models Using Event Playback Approach", IEEE Access, vol. 10, no. 3, 2022, pp. 72132-72144.
[13]. *R. Huang, R. Diao, Y. Li, J. Sanchez-Gasca, Z. Huang* and *B. Thomas,* "Calibrating parameters of power system stability models using advanced ensemble Kalman filter", IEEE Transactions on Power Systems, vol. 33, no. 3, 2018, pp. 2895-2905.
[14]. *R. Ehsan, F. Jawad, A, Moein,* "Detection of Loss-of-Excitation in Synchronous Machines using Hybrid GMDH Neural Network Intelligent Algorithm in the Presence of UPFC in Power System", Electric Power Components and Systems, vol. 36, no. 6, 2023, pp. 972-990.
[15]. *S. Wang, R. Diao, T. Lan, Z. Wang* and *X. Lu*, "A DRL-aided multi-layer stability model calibration platform considering multiple events", in Proc. 2020 IEEE Power & Energy Society General Meeting (PESGM), Montreal, QC, Canada, 2020, pp. 1-5.
[16]. *S. Badakhshan, A. R. Jacob, B. Li, et al.,* "Self-healing power systems using reinforcement learning over graphs for controlled grid islanding", Sustainable Energy, Grids and Networks, vol. 44, no. 9, 2025, pp. 101937-101937.
[17]. *A. Fathollahi,* "Machine Learning and Artificial Intelligence Techniques in Smart Grids Stability Analysis: A Review", Energies, vol. 18, no. 13, 2025, pp. 3431-3431.
[18]. *Y. Zhang, J. Liu, F. Qiu, T. Hong* and *R. Yao*, "Deep active learning for solvability prediction in power systems", Journal of Modern Power Systems and Clean Energy, vol. 10, no. 6, 2022, pp. 1773-1777.
[19]. *M. Mishra, G. J. Singh,* "A comprehensive review on deep learning techniques in power system protection: Trends, challenges, applications and future directions", Results in Engineering, vol. 19, no. 1, 2025, pp. 25103884-103884.
[20]. *H. Ma, X. Lei, Z. Li, S. Yu, B. Liu* and *X. Dong*, "Deep-learning-based power system events detection technology using spatio-temporal and frequency information", IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 13, no. 2, 2023, pp. 545-556.
[21]. *D. Kim, G. Ryu, C. Moon* and *B. Kim*, "Accuracy of a short-term wind power forecasting model based on deep learning using LiDAR-SCADA integration: A case study of the 400-MW Anholt offshore wind farm", Applied Energy, vol. 373, 2024, p. 373123882-123882.