Practical Methods of Defective Input Feature Correction to Enable Machine Learning in Power Systems

Jingzi Liu, *Graduate Student Member, IEEE*, Fangxing Li, *Fellow, IEEE*, Francisco Zelaya-Arrazabal, *Graduate Student Member, IEEE*, Hector Pulgar-Painemal, *Senior Member, IEEE*, and Hongyu Li, *Member, IEEE*

***Abstract*—In this research work, three practical correction methods are proposed to mitigate the impact of defective input features in power system data measurement for machine learning (ML) applications. A well-trained ML tool may become ineffective due to defective input features, which may originate from measurement issues, such as monitor malfunction, cyberattack, communication failure, or others. It is crucial to correct defective input features to enable ML tools with desirable performances. This letter first introduces the mechanism of three correction methods, i.e., statistical-value-based method, minimal-error-based method, and DNN-based adaptive method. Then, the methods are validated via a deep neural network (DNN) case for power system stability enhancement. Validation results demonstrate that the adaptive method achieves the best performance, enabling the well-trained ML tool with a similar accuracy level to the case of no data defects. Although actual measurements may have various data issues challenging ML applications in power systems, the proposed methods show promise for addressing these challenges.**

***Index Terms*—Defective data, data correction, data preprocessing, measurement, machine learning, power system.**

I. Introduction

IN RECENT years, the potential of machine learning (ML) in enhancing power system operation and planning has gained significant attention due to its powerful computing, analytical, and processing capabilities [1]. Typically, an ML-based model undergoes training followed by its online application with online-measured data. Only when normal input data is fed into the trained ML model, can this process run smoothly. However, it is not easy to consistently measure normal data in power systems because there are risks of typical defects like data loss, data latency, etc., from power system data measurement [2], especially suffering from monitor malfunction, communication failure, cyberattack, etc. As such, it is worth exploring how to enable a well-trained ML tool to function reliably even with defective input data. This topic holds particular importance in the context of ML applications in power systems, where defects in power system measurement tend to arise.

Manuscript received 9 June 2023; revised 11 September 2023; accepted 16 October 2023. Date of publication 27 October 2023; date of current version 26 December 2023. This work was supported by the NSF under Grant ECCS-2033910. Paper no. PESL-00180-2023. *(Corresponding author: Fangxing Li.)*

The authors are with the University of Tennessee, Knoxville, TN 37996 USA (e-mail: jliu104@vols.utk.edu; fli6@utk.edu; fzelayaa@vols.utk.edu; hpulgar@utk.edu; hli90@utk.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TPWRS.2023.3328161.

Digital Object Identifier 10.1109/TPWRS.2023.3328161

The most direct solution is to correct defects. Generative adversarial network, extreme learning machine, and random vector functional link networks are cooperated to fill up phasor measurement unit (PMU) missing data [3]. PMU data is arranged in three dimensions by polyadic tensor decomposition and Tucker tensor decomposition to recover missing data [4]. Both [5] and [6] recover missing PMU data by alternative direction method of multipliers, but the work is formulated as a low-rank matrix-completion problem [5] and an estimation by decomposing events in non-dynamic and dynamic elements [6].

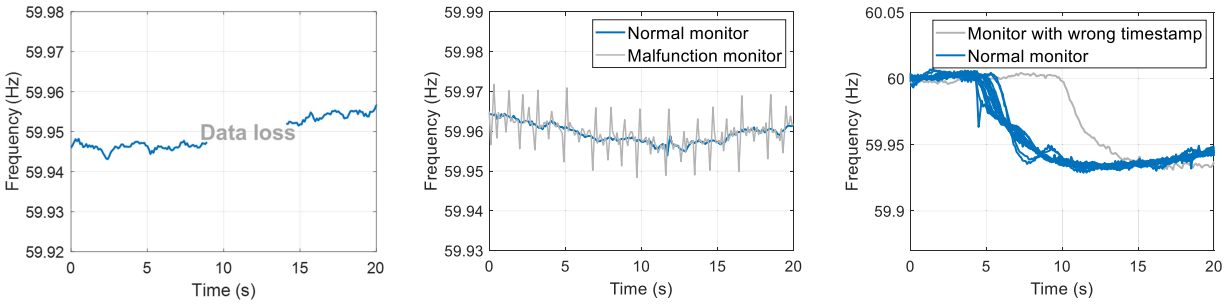
To provide practical methods that are straightforward, yet effective to implement, three data-driven correction methods are proposed in this letter to enable ML in power systems.

II. Research Motivation

In practical applications, the data used by ML tools in power systems, whether for training or online applications, is usually from power system data measurement systems, like PMUs or Supervisory Control and Data Acquisition (SCADA) systems. Normally, measurements are complete and accurate. However, due to various issues, measured data for ML may be defective, leading to failed ML results. Fig. 1 depicts three common PMU data defects: data loss, bad data, and data asynchronization.

To indicate the solutions for defective data, an ML-based power system stability enhancement case is introduced. Prior to the online application, this ML case is trained by historical data first. The input data employed to construct and train the model is sourced from PMUs, encompassing PMU frequency and tie-line power flow measurements. The output data is an index indicating the status of power system stability. During its online application, this well-trained ML tool can assess stability with normal input data. However, as shown in Fig. 2, this trained ML tool may not be effective with defective input features depicted in Fig. 1. Even if only one input feature is defective, the well-trained ML tool may become less accurate in its performance.

It is crucial to correct defective input features to enable a well-trained ML tool to function with desirable performances. To address these real-world issues, this letter proposes three practical correction methods for defective input features. In essence, each correction method functions as an additional module during ML application, as the blue block shown in Fig. 3. The detailed procedures are elaborated in the next sections.



(a) PMU data loss (b) Bad PMU data (c)PMU data asynchronization

Fig. 1. Power systems measurement defective data issues.

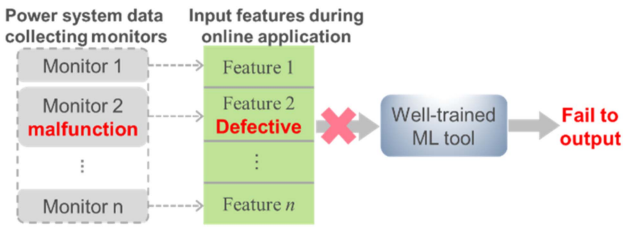


Fig. 2. Motivation of the proposed research work.

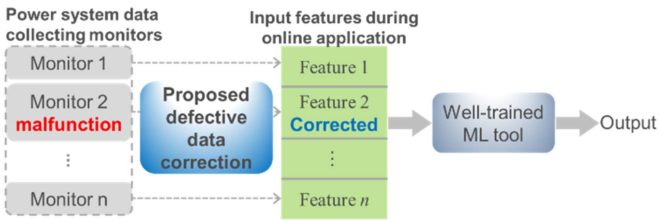


Fig. 3. Idea of defective input feature correction.

III. Three Correction Methods

This letter proposes three practical data-driven methods to correct defective input features in power system ML applications. The detailed descriptions of the mechanism employed by each method are elaborated.

A. Statistical-Value-Based Method

The first algorithm relies on statistical value to correct defective input features. Common statistical values to describe the central tendency of a dataset, such as the mean and median, are used here to correct defective input features. As illustrated in Fig. 4, in the instance where collected values of the second input feature (i.e., Feature 2) during the online application are defective, the mean or median of the same-type input feature (i.e., also in the second position within the historical training set of b samples) $\{F_{21}, F_{22}, \ldots, F_{2b}\}$ is used to complement defects, as shown by (1) or (2), respectively.

$$\text{Corrected Feature 2} = \text{mean}\{F_{21}, \ldots, F_{2b}\} \quad (1)$$

$$\text{Corrected Feature 2} = \text{median}\{F_{21}, \ldots, F_{2b}\} \quad (2)$$

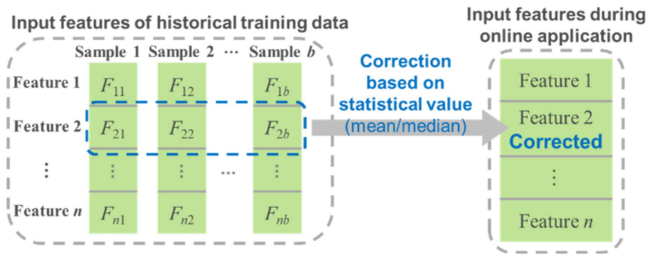


Fig. 4. Process of statistical-value-based method.

B. Minimal-Error-Based Method

The second algorithm aims to derive the value with minimal prediction error from the known entries of the historical training input data and the well-trained ML tool. The essence lies in providing an initial assessment by the well-trained ML tool to yield the value with minimal prediction error, identifying it as the most suitable option for correcting defective input features. For further clarity, the entire process is introduced in Fig. 5.

In the same data case as the previous method, the values of Feature 2 are defective for the online application. The minimum and maximum values of the same-type input feature $\{F_{21}, \ldots, F_{2b}\}$ within the historical training set are identified first. Then, m independent and identically distributed (i.i.d.) random numbers, $V_1, V_2, \ldots, V_m$, are generated over the range spanning from the minimum to maximum values of uniform distribution. Subsequently, V_1 is initially chosen to replace F_{21} to F_{2b}, thus forming a new set of input data. Following this, a prediction error of this new input dataset is evaluated by the known trained ML tool. Finally, with the adoption of a sequential substitution strategy, V_2 through V_m are successively employed to replace $\{F_{21}, \ldots, F_{2b}\}$ and generate new input sets. Each new input set is assessed for its respective prediction error via the known well-trained ML tool. Upon comparison, the specific V_i with the minimal prediction error is selected to complement defects.

C. DNN-Based Adaptive Method

An adaptive algorithm is devised with the core of leveraging an adaptive tool to capture features among historical input data,

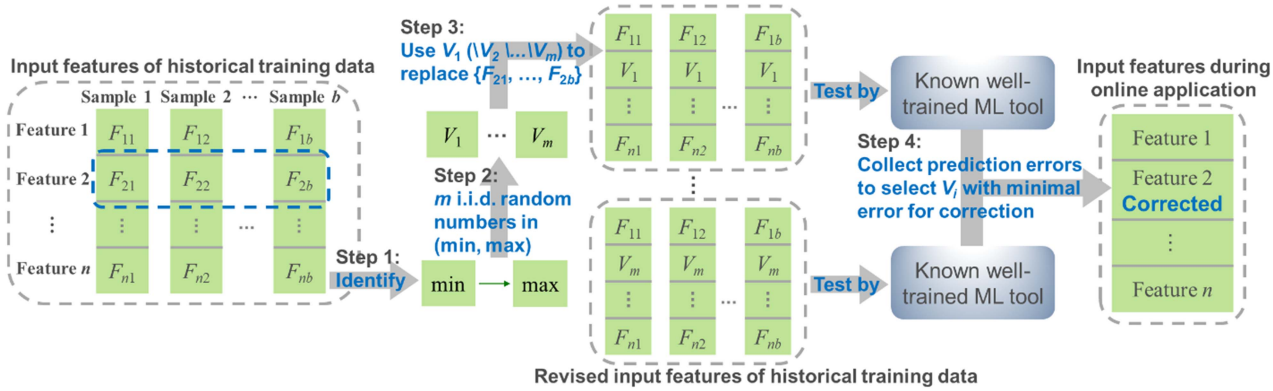


Fig. 5. Process of minimal-error-based method.

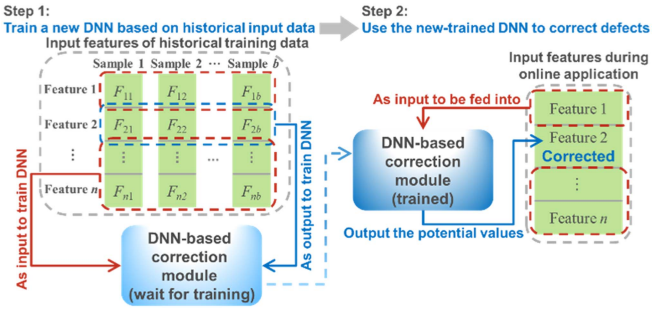


Fig. 6. Process of DNN-based adaptive method.

then yielding potential values for defective input features. As depicted in Fig. 6, deep neural network (DNN) is chosen due to its robust capability of replicating intrinsic data characteristics in a straightforward and precise manner, under multi-hidden-layered configuration and potent fitting capability of parameters (neurons, weights, bias, etc.) in each layer, distinguishing it from alternative adaptive tools.

Similarly, in the example of a defect in Feature 2 during the online application, input features within the historical training set are retrieved first. Then, a DNN model is trained using the outputs of $\{F_{21}, \ldots, F_{2b}\}$ and the inputs of the remaining historical training set excluding $\{F_{21}, \ldots, F_{2b}\}$. Subsequently, the recent input features from the online application excluding Feature 2 served as the input variable for the newly trained DNN model to yield potential values for the defective Feature 2. Following this step, each defective input feature can have its own dedicated DNN-based correction model. As such, DNN correction models can be pre-trained and stored in memory. When certain defective input features are detected during the online application, the corresponding DNN-based correction model can be quickly invoked to implement the correction.

IV. Case Study

A DNN-based power system stability enhancement case is introduced as a benchmark, which also acts as the validation case of no input date issues. The data is simulated from PowerFactory on a reduced 179-bus model for Western Electricity Coordinating Council (WECC), which includes multiple snapshots of the grid situation under various operating conditions and disturbances. Gaussian noise is embedded in this dataset to better represent real-world data. More details can be found in [7]. 90% of this dataset, called as training set, is randomly selected to train the DNN case, while the remaining 10%, known as validation set, is reserved for assessing the performance of the proposed methods.

Given that the validation set contains 41 input features, each of 41 features is independently and sequentially treated as a defective case for an efficient and comprehensive validation. Consequently, each proposed method is validated by 41 sets of tests. First, three methods are successively used with the well-trained DNN case to yield 41 outputs of test sets. Subsequently, each set of output from the same method is evaluated against the ground truth (i.e., output data from the validation set with no defective issues) by the mean error rate (MER) outlined in (3). Finally, the MER of each method for each input defect within 41 tests among multiple data samples in the validation set is computed to assess correction performances.

$$\mathrm{MER}\,(j) = \left(\sum_{i=1}^{N} \frac{|Y_{\mathrm{corrected}}\,(i) - Y_{\mathrm{truth}}\,(i)|}{Y_{\mathrm{truth}}\,(i)} \Big/ N \right) \times 100\% \tag{3}$$

where $Y_{\mathrm{corrected}}$ denotes the output data of the trained DNN case using corrected input data; Y_{truth} is the output data of the trained DNN case using input data with no defects. N is the total number of samples in the validation set, $i = 1, \ldots, N$. $j = 1, \ldots, 41$.

In addition to the metric of accuracy, computational time is also a crucial metric. Table I presents a comparison of each method's average computational time of 41 sets of tests, as well as Fig. 7 presents a comparison of all cases' accuracy. Except for the 3rd, 19th, and 28th input features with a higher sensitivity for input data defects, MER values for the rest input features are all below 10%. The statistical-value-based method (orange star and green cross) shows the highest MER but offers the most straightforward and time-saving application process. The minimal-error-based method (blue triangle) has a slightly lower MER compared to the previous method and the fastest correction process. However, a significant amount of time is required during

TABLE I
COMPARISON OF COMPUTATIONAL TIME OF EACH METHOD

Method		Computational time/s	
		Preparation	Correction
Statistical-value-based method	Mean	N/A	0.0009
	Median	N/A	0.0146
Minimal-error-based method		444.7834	0.0001
		444.7835 (total)	
DNN-based adaptive method		11.0339	0.1222
		11.1561 (total)	

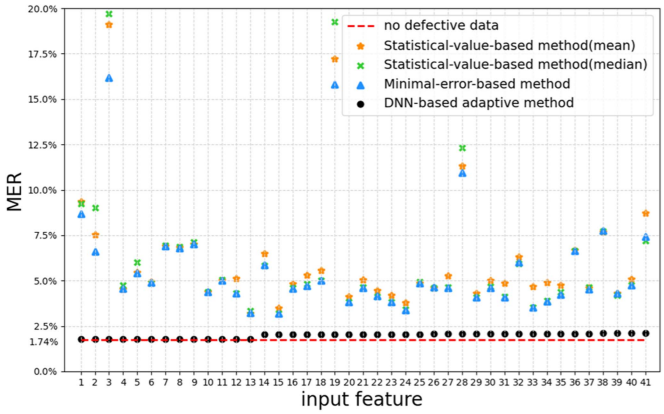


Fig. 7. Comparison of MER of each case.

its preparation to find the suitable value with minimal error. The DNN-based adaptive method (black dot) exhibits the lowest MER and acceptable time, which even enables the well-trained ML tool to attain nearly the same accuracy level as the case of no defective data.

V. CONCLUSION

In power system ML applications, defective measurement data issues may arise, particularly in incidents like monitor malfunctions, cyberattacks, communication failures, etc., and thus the well-trained ML tools may fail. Hence, it is essential to correct these defective input data for the effectiveness of well-trained ML tools. This letter proposes three easy-to-implement methods to maximize the utilization of available resources to correct defective input features in real power system scenarios.

The statistical-value-based method works only by statistical values of historical data, making correction straightforward. This results in the most time-saving application process, but it comes with low accuracy. If accuracy tolerance is not high, the statistical-value-based method can be considered. The minimal-error-based method enhances accuracy, but the preparation process is time-consuming and complex, which also affects the overall application speed. The DNN-based adaptive method demonstrates the best correction performance in that the preparation and correction processes are comparatively less time-consuming, nearly reaching the same accuracy level compared to it from no data defects.

Given that this research's motivation is to correct defective input features, the location information about defective features is known to the proposed methods. However, such location information may not be available in some defects, such as bias, drift, and freeze faults [8], [9]. Thus, it might be necessary to build a defective data detection before correction to identify and locate defects, which is worthy of future study. Furthermore, the proposed methods conduct correction based on historical training data, so the relevance between online application data and historical data plays a key role in providing the desirable performance. As such, research for the scenario of less relevance between online application data and historical data is also worth exploring in the future. Meanwhile, the impact of the data amount and locations on the correction performance is also an interesting future topic. In addition, correction methods by ML approaches beyond DNN can be investigated in the future.

REFERENCES

[1] Q. Zhang, F. Li, W. Feng, X. Wang, L. Bai, and R. Bo, "Building marginal pattern library with unbiased training dataset for enhancing model-free load-ED mapping," *IEEE Open Access J. Power Energy*, vol. 9, pp. 88–98, 2022.
[2] C. Huang et al., "Data quality issues for synchrophasor applications part II: Problem formulation and potential solutions," *J. Modern Power Syst. Clean Energy*, vol. 4, no. 3, pp. 353–361, Jul. 2016.
[3] C. Ren and Y. Xu, "A fully data-driven method based on generative adversarial networks for power system dynamic security assessment with missing data," *IEEE Trans. Power Syst.*, vol. 34, no. 6, pp. 5044–5052, Nov. 2019.
[4] D. Osipov and J. H. Chow, "PMU missing data recovery using tensor decomposition," *IEEE Trans. Power Syst.*, vol. 35, no. 6, pp. 4554–4563, Nov. 2020.
[5] M. Liao, D. Shi, Z. Yu, Z. Yi, Z. Wang, and Y. Xiang, "An alternating direction method of multipliers based approach for PMU data recovery," *IEEE Trans. Smart Grid*, vol. 10, no. 4, pp. 4554–4565, Jul. 2019.
[6] B. Foggo and N. Yu, "Online PMU missing value replacement via event-participation decomposition," *IEEE Trans. Power Syst.*, vol. 37, no. 1, pp. 488–496, Jan. 2022.
[7] F. Zelaya-Arrazabal, J. Liu, J. Zhao, H. Pulgar-Painemal, F. Li, and H. Silva-Saravia, "Data-driven adaptive dynamic coordination of damping controllers," in *Proc. North Amer. Power Symp.*, 2022, pp. 1–6.
[8] H. Darvishi, D. Ciuonzo, and P. S. Rossi, "A machine-learning architecture for sensor fault detection, isolation, and accommodation in digital twins," *IEEE Sensors J.*, vol. 23, no. 3, pp. 2522–2538, Feb. 2023.
[9] Y. Li and X. Shen, "A novel wind speed-sensing methodology for wind turbines based on digital twin technology," *IEEE Trans. Instrum. Meas.*, vol. 71, 2022, Art. no. 2503213.