07 Feb 2022

# Building Marginal Pattern Library with Unbiased Training Dataset for Enhancing Model-Free Load-ED Mapping

Qiwei Zhang

Fangxing Li

Wei Feng

Xiaofei Wang

*et. al. For a complete list of authors, see https://scholarsmine.mst.edu/ele_comeng_facwork/4552*



## Recommended Citation

Q. Zhang et al., "Building Marginal Pattern Library with Unbiased Training Dataset for Enhancing Model-Free Load-ED Mapping," *IEEE Open Access Journal of Power and Energy*, Institute of Electrical and Electronics Engineers (IEEE), Feb 2022.
The definitive version is available at https://doi.org/10.1109/OAJPE.2022.3149308

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/OAJPE.2022.3149308, IEEE Open Access Journal of Power and Energy

# Building Marginal Pattern Library with Unbiased Training Dataset for Enhancing Model-Free Load-ED Mapping

Qiwei Zhang, *Student Member, IEEE*, Fangxing Li, *Fellow, IEEE,* Wei Feng, *Member, IEEE*, Xiaofei Wang, *Student Member, IEEE,* Linquan Bai, *Senior Member, IEEE*, Rui Bo, *Senior Member, IEEE*

***Abstract*— Input-output mapping for a given power system problem, such as loads versus economic dispatch (ED) results, has been demonstrated to be learnable through artificial intelligence (AI) techniques, including neural networks. However, the process of identifying and constructing a comprehensive dataset for the training of such input-output mapping remains a challenge to be solved. Conventionally, load samples are generated by a pre-defined distribution, and then ED is solved based on those load samples to form training datasets, but this paper demonstrates that such dataset generation is biased regarding load-ED mapping. The marginal unit and line congestion (i.e., marginal pattern) exhibit a unique characteristic called "step change" in which the marginal pattern changes when the load goes from one critical loading level (CLL) to another, and there is no change of marginal units within the interval of the two adjacent CLLs. Those loading intervals differ significantly in size. The randomly generated training dataset overfills intervals with large sizes and underfits intervals with small sizes, so it is biased. In this paper, three algorithms are proposed to construct a marginal pattern library to examine this bias according to different computational needs, and an enhancement algorithm is proposed to eliminate the bias for the load-ED dataset generation. Three illustrative test cases demonstrate the proposed algorithms, and comparative studies are constructed to show the superiority of the enhanced, unbiased dataset.**

***Index Terms*—Neural networks, Critical load level (CLL), Locational marginal price (LMP), Economic dispatch (ED), Optimal power flow (OPF), Electricity market.**

## NOMENCLATURE

*Sets and Indices*

| | |
|---|---|
| $N_G$, $N_b$ | Number of generators and number of buses |
| $N_l$ | Number of lines |
| $N_{co}$ | Number of marginal pattern combinations |
| $CL$, $UL$ | Set of congested and uncongested lines |
| $MG$, $NG$ | Set of marginal units and non-marginal units |
| $L^P$, $L^N$ | Set of positive congested lines and negative congested lines |

*Parameters*

| | |
|---|---|
| $C_i$ | Cost of unit $i$ |
| $DF_i$ | Delivery factor of unit $i$ |
| $GSF_{l\text{-}i}$ | Generation shift factor of bus $i$ to line $l$ |
| $P_i^{max}$, $P_i^{min}$ | Upper and lower generation capacity limits of unit $i$ |
| $L_l^{max}$, $L_l^{min}$ | Upper and lower transmission thermal limits of line $l$ |
| $\varepsilon$ | Small positive number, such as 0.01 |
| $\rho$, $\sigma$ | User-defined value for perturbations of variable $\beta_i$ |
| $\omega^{max}$, $\omega^{min}$ | User-defined upper and lower limits for variable $\beta_i$ |

*Variables*

| | |
|---|---|
| $P_i$, $\Delta P_i$ | Power output of unit $i$ and incremental power output of unit $i$ |
| $FL_l$, $\Delta FL_l$ | Line flow at line $l$ and incremental line flow at line $l$ |
| $MP$ | Power output of marginal units |
| $S_{CL}$, $S_{UL}$ | Slack variables for congested and uncongested line constraints |
| $d_i$, $\Delta d_i$ | Load at bus $i$ and incremental load at bus $i$ |
| $\Delta D_\Sigma$ | Incremental loading level |
| $LMP_i$ | Locational marginal price at bus $i$ |
| $\beta_i$ | Incremental variable for the iterative process in Algorithm DE |
| $\lambda$ | Lagrangian multiplier for power balance constraint |
| $\mu_l^+$, $\mu_l^-$ | Lagrangian multiplier for $l^{th}$ upper and lower line flow limits |
| $\eta_i^+$, $\eta_i^-$ | Lagrangian multiplier for $i^{th}$ upper and lower generator output limits |

## I. INTRODUCTION

THE 2016 victory of AlphaGo, a computer program that defeated the strongest human Go player in the world, demonstrated the potential of artificial intelligence (AI) for solving complex decision-making problems [1]. The continuous development of AI is profoundly impacting everyday life and industrial developments. The increasing digitalization of the power grid and impressive leaps in computation capabilities are unlocking the possibility of AI-enhanced power systems [2]. The speed of development of AI techniques is revolutionizing traditional power system planning

Q. Zhang, F. Li, W. Feng, and X. Wang are with the Min H. Kao Department of EECS, The University of Tennessee, Knoxville, TN 37996 USA (e-mail: fli6@utk.edu).

L. Bai is with the Systems Engineering and Engineering Management Department, University of North Carolina at Charlotte, Charlotte, NC 28223 USA.

Rui Bo is with the Department of ECE, Missouri University of Science and Technology, Rolla, MO 65409 USA.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/OAJPE.2022.3149308, IEEE Open Access Journal of Power and Energy



and operations. Ref. [3] compared power system AI with the epic AlphaGo computer program and sketched promising prospects of implementing AI techniques in the power system. Ref. [4] analyzed the opportunities and challenges of adapting and developing AI techniques in transmission, distribution, microgrids, and multi-energy systems.

One of the most recent applications of an AI technique in power systems is the use of AI in economic dispatch (ED), which is essentially a security-constrained optimal power flow (OPF) problem. Note, once the ED is solved, some results, such as generation dispatches, unsupplied loads, and system total cost, are directly available. Meanwhile, indirect results such as reliability indices and locational marginal prices (LMPs) can be easily obtained.

The ED problem must be solved repetitively within a short time during daily operations. Therefore, recent research has attempted to directly predict the results of the OPF-based ED problem through neural networks without solving the optimization model. In ref. [5], various regression models, including a support vector regression model and a fully connected neural network, were applied to predict optimal dispatch results based on load and contingency data. Similarly, ref. [6] built a neural network to learn the mapping between the load and generation dispatches. In ref. [7], a graph neural network was constructed to predict the optimal dispatch results based on loads. In ref. [8], a deep neural network was combined with the Lagrangian dual method to improve the accuracy of the prediction of optimal dispatch results.

Most recently, ref. [9] developed the DeepOPF approach, which applies a deep neural network to predict the dispatch result of a linearized OPF problem. With a linearized power flow, the OPF-based ED problem becomes convex. Although solving the linearized ED problem is generally efficient, it incurs computational complexity depending on different applications [10][11]. Some of the literature has applied data-driven learning techniques to identify specific patterns of the linearized ED problem instead of directly predicting ED outputs. In ref. [12], a neural network was proposed to predict the umbrella constraints that form the feasibility regions of the ED problem. In ref. [13], a neural network classifier was proposed to learn the binding constraint of the ED problem. In ref. [14], statistic learning was applied to learn the mapping between the optimal basis of ED and uncertainties.

In summary, previous research has applied different types of neural networks to predict the following four outputs, direct or indirect, of the ED problem: (1) the optimal dispatch results; (2) the optimal cost; (3) the reliability index/status; and (4) the characteristic of the optimal solutions. However, the dataset (i.e., load versus above four outputs) is generally produced by a pre-defined distribution without considering the intrinsic characteristic of the ED problem. For example, research works [8] and [9] apply uniform distribution to generate load samples. Research works [13] and [14] use normal distribution to generate load samples. In general, the most straightforward way to generate a large set of load samples is to directly apply a certain distribution. However, in this paper, we demonstrate that the randomly generated load samples are biased in relation to ED outputs, such as generator dispatches and LMPs. Here, three algorithms are proposed to construct the marginal pattern library, and another algorithm is proposed to enhance the dataset for model-free applications in ED and LMP calculations. In summary, this paper provides a better way to generate load samples for the training dataset of load-ED mapping.

The main contributions of this paper are two-fold:

- This work identifies that a randomly generated dataset is biased for ED output prediction, even when the dataset capacity is large. The loading intervals for different marginal patterns differ significantly in size. It is possible that the randomly generated datasets overfill the intervals with large sizes and underfit the intervals with small sizes. It is worth noting that a loading interval with a large size is not necessarily more important than a loading interval with a small size. It is possible that load vs. ED outputs may vary considerably in a small loading interval for an intra-day operation, so it is important to understand the behavior of load vs. ED in this small interval.
- This work proposes three algorithms to construct a marginal pattern library and examine the dataset: (1) a comprehensive enumeration algorithm; (2) an iterative search algorithm; and (3) a fast screening algorithm. An enhancement algorithm is also proposed to enhance the training dataset according to the marginal pattern library. The characteristics of the proposed algorithms are illustrated with several examples, and a comparative study demonstrates the effectiveness of the enhancement algorithm.

The rest of this paper is organized as follows. Section II briefly reviews the formulations of the ED problem. Section III discusses the phenomenon that a randomly generated dataset is biased for predicting ED-based problems. In Section IV, the three proposed algorithms for marginal pattern collection and the dataset enhancement algorithm are presented with illustrative cases. Section V describes a comparative study demonstrating the effectiveness of the enhanced dataset. Finally, a conclusion is drawn in Section VI.

## II. Preliminaries on ED and LMPs

The ED problem is typically formulated as a linearized OPF problem in most of the ISOs due to the computation issue [15]. A general ED problem with line flow limits and unit capacity constraints is formed in (1a)-(1d). Other technical constraints, such as N-1 contingency scenarios and reserves, could be added by modifying the constraint set in problem (1) but they are not explicitly modeled here for illustration simplicity.

$$\min \sum_{i}^{N_G} C_i P_i \tag{1a}$$

$$\sum_{i} DF_i \times P_i - \sum_{i} DF_i \times d_i = 0 : \lambda \tag{1b}$$

$$P^{\min}{}_i \le P_i \le P^{\max}{}_i, \forall i \in N_b : \eta^-, \eta^+ \tag{1c}$$

$$L_l^{\min} \le \sum_{i=1}^{N_b} GSF_{l-i}(P_i - d_i) \le L_l^{\max} \forall l \in N_L : \mu^+, \mu^- \tag{1d}$$

The LMP is calculated after the ED result in (1a)-(1d) is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/OAJPE.2022.3149308, IEEE Open Access Journal of Power and Energy



obtained, so it is an indirect result of the ED problem. The LMP pricing scheme has been widely adopted in U.S. electricity markets to provide economic signals to market participants. LMPs are defined as the marginal increase in dispatch costs versus the marginal increase in load consumption at a particular bus, as given in (2) [16].

$$LMP_i = \lambda + \sum_{i}^{N_B} GSF_{l-i}(\mu^- - \mu^+) + (1 - DF_i)\lambda \quad (2)$$

## III. The Biased Dataset for Data-Driven ED Model

The ED results, such as generation dispatches or LMPs, have a unique characteristic called "step change" [17] at some specific system load levels. The load level at which a step change of LMP occurs is referred to as a critical load level (CLL). As discussed in [17], when the loading level varies within a certain interval, the marginal unit output and load flow also vary w.r.t. the loads according to a certain pattern. In this paper, loading level refers to the sum of loads in the system or system loading level. This phenomenon is also identified as "system pattern regions" in [18] and [19]. When the loading level steps out of the interval, the pattern changes instantaneously with a step change of LMP [20].

If the load at each bus can be grouped by a set of participating factors, then each marginal pattern corresponds to a continuous loading interval. If the load at each bus changes individually, then each marginal pattern corresponds to a multi-dimensional region (also referred to as loading interval in this paper). Under either situation, some of the marginal patterns correspond to large loading intervals, while the others correspond to small loading intervals. Fig. 1 shows an illustrative example of marginal patterns and various CLLs where price step changes occur. The loading intervals between two adjacent CLLs for MP1, MP5, and MP6 are larger than the loading intervals for MP2, MP3, and MP4. Below are some crucial observations of marginal patterns (MPs) and loading levels in Fig. 1:

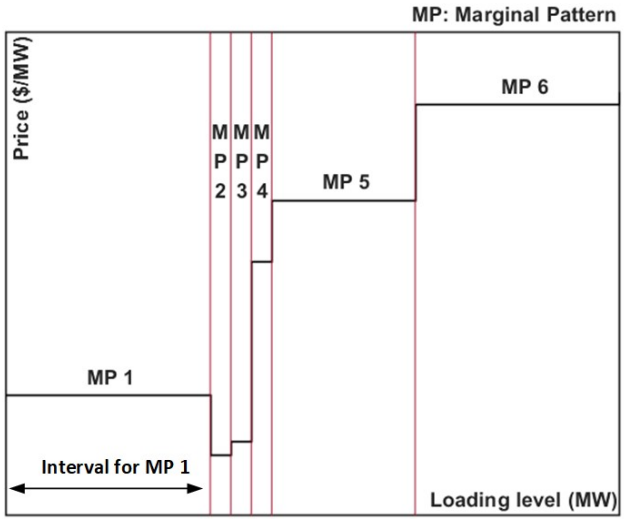


Fig. 1. Marginal patterns at three large intervals (MP1, MP5, and MP6) and three small intervals (MP2, MP3, and MP4)

- If load samples are generated according to pre-defined distributions (e.g., uniform, normal or Weibull distribution), the marginal pattern of a large interval will have many more training samples than the marginal pattern of a small interval. For example, if the dataset is generated randomly according to a uniform distribution, the probability of samples landing on different marginal patterns is shown in Table I, where the percentages are rounded to the nearest integers. Most of the training samples will locate in MP1, MP5, and MP6, and only a small portion of the samples will locate in MP2, MP3, and MP4. Seemingly, this is reasonable because large intervals have more training samples. However, this may lead to insufficient number of training samples in a small interval to have good results. In plain language, a large interval may have an unnecessarily large number of training samples, while a small interval may not have enough training samples – possibly 0 samples in an extreme case.
- Note, a small interval is not necessarily less important than a large interval. It is possible that load vs. ED outputs may vary considerably in a small loading interval for an intra-day operation, so it is important to understand the behavior of load vs. ED in this small interval.
- Ideally, the number of training samples across all marginal patterns should be sufficiently large. It is preferred that the number of training samples is the same in each interval, rather than based on the width of intervals.

Table I. Probability of a load sample falling within marginal patterns

| Pattern | MP 1 | MP 2 | MP 3 | MP 4 | MP 5 | MP 6 |
|---|---|---|---|---|---|---|
| Probability | 33% | 3% | 3% | 3% | 23% | 33% |

In summary, if the training dataset and test dataset are generated together by a pre-determined distribution, then the test dataset also contains few test samples in the marginal pattern with small sizes, making the data-driven prediction less accurate. In other words, the biased training dataset eventually leads to a biased neural network. A detailed example is provided in Section V for comparative case studies.

## IV. Datasets Examination and Enhancement Algorithms

The phenomenon of dataset absence in a marginal pattern with small loading intervals calls for efficient examination and enhancement methods. This section consists of two parts: (1) three algorithms are proposed to collect marginal patterns, which construct a marginal pattern library examining the training dataset; (2) if the marginal pattern library contains patterns that are missing in the training dataset, an enhancement algorithm is proposed to eliminate the bias in samples by generating samples for those marginal patterns.

### *A. Constructing marginal patterns library*

*Algorithm CE: comprehensive enumerations*

The optimality of the optimization problem (1) always lies in the extrema of the constraint set [23], namely the intercepts of binding constraints. Therefore, solving problem (1) is equal to solving a system of linear equations, which means the optimum of problem (1) can be represented by (3) in a matrix representation. The generation variables are divided into the generation of marginal unit *MG* and the generation of non-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/OAJPE.2022.3149308, IEEE Open Access Journal of Power and Energy



Table II. Marginal patterns for PJM-5 bus system in Algorithm 1

| | Pattern 1 | Pattern 2 | Pattern 3 | Pattern 4 | Pattern 5 | Pattern 6 | Pattern 7 |
|---|---|---|---|---|---|---|---|
| Bus 1 ($) | 10 | 14 | 14 | 12.13 | 15 | 14 | 14 |
| Bus 2 ($) | 10 | 14 | 15 | 15 | 15 | 19.39 | 15 |
| Bus 3 ($) | 10 | 14 | 14.81 | 16.10 | 15 | 21.47 | 15.38 |
| Bus 4 ($) | 10 | 14 | 14.29 | 19.14 | 15 | 27.17 | 16.44 |
| Bus 5 ($) | 10 | 14 | 14.05 | 10 | 15 | 10 | 13.26 |

| | Pattern 8 | Pattern 9 | Pattern 10 | Pattern 11 | Pattern 12 | Pattern 13 | Pattern 14 | Pattern 15 |
|---|---|---|---|---|---|---|---|---|
| Bus 1 ($) | 16.99 | 16.98 | 30 | 14 | 8.65 | 40 | 14 | 3.54 |
| Bus 2 ($) | 26.42 | 26.38 | 30 | 33.74 | 34.99 | 40 | 103.81 | 129.49 |
| Bus 3 ($) | 30.04 | 30 | 30 | 30 | 30 | 40 | 86.79 | 105.63 |
| Bus 4 ($) | 40 | 39.94 | 30 | 19.71 | 16.27 | 40 | 40 | 40 |
| Bus 5 ($) | 10 | 10 | 30 | 15.01 | 10 | 40 | 18.61 | 10 |

marginal unit *NG*. The binding and non-binding line flow constraints are indicated by *CL* and *UL,* respectively. Equation (3) holds for any solution to problem (1).

$$\begin{bmatrix} 1 & 0 \\ GSF_{CL,MG} & 0 \\ GSF_{UL,MG} & I \end{bmatrix} \times \begin{bmatrix} P_{MG} \\ S_{UL} \end{bmatrix} = \begin{bmatrix} 0 \\ L_{CL}^{\max} \\ L_{UL}^{\max} \end{bmatrix} + \begin{bmatrix} 1 \\ GSF_{CL,d} \\ GSF_{UL,d} \end{bmatrix} \times \begin{bmatrix} d_1 \\ \dots \\ d_{N_B} \end{bmatrix} + \begin{bmatrix} -1 & 0 \\ -GSF_{CL,NG} & I \\ -GSF_{UL,NG} & 0 \end{bmatrix} \times \begin{bmatrix} P_{NG} \\ S_{CL} \end{bmatrix} \tag{3}$$

Under a given network, the value of *GSF* is constant. Thus, a sensitivity matrix **W** of loads and basic variables (i.e., $P_{MG}$ and $S_{UL}$) can be obtained as in (4), which represents the change of marginal unit output and the line flow change in uncongested lines if there is a load increase at a particular bus.

$$\begin{bmatrix} \Delta P_{MG} \\ \Delta S_{UL} \end{bmatrix} = \mathbf{W} \times \begin{bmatrix} \Delta d_1 \\ \dots \\ \Delta d_{N_B} \end{bmatrix} \text{ where}$$

$$\mathbf{W} = \begin{bmatrix} 1 & 0 \\ GSF_{CL,MG} & 0 \\ GSF_{UL,MG} & I \end{bmatrix}^{-1} \times \begin{bmatrix} 1 \\ GSF_{CL,d} \\ GSF_{UL,d} \end{bmatrix} \tag{4}$$

The selection of *MG* and *UL* in (4) determines the marginal pattern. A marginal pattern is uniquely labeled by LMPs, as shown in (5). Each marginal pattern corresponds to a CLL. It is worth noting that outputs of the ED problem, such as optimal dispatches, marginal patterns, and LMPs, are consistent, which means identifying one of them is equivalent to identifying all [21]. The following discussion will focus on identifying LMPs.

$$LMP = f(MG, CL) \tag{5}$$

If the binding constraints in the market-clearing model (1) are determined, the LMP is also determined. LMPs can be represented as the cost of serving the next incremental load that is covered by the marginal units, as shown in (6).

$$LMP_i = \frac{\partial \sum_i^{N_G} C_i(P_i)}{\partial d_i} = \sum_i^{MG} C_i \frac{\partial P_i}{\partial d_i} \tag{6}$$

By substituting (4) into (6), LMPs can be formulated as in (7), where matrix $\mathbf{W_{MG}}$ represents the row that corresponds to marginal units in matrix **W**. The values of matrix $\mathbf{W_{MG}}$ are determined by the set of marginal units and congested lines.

$$\begin{bmatrix} LMP_1 \\ \dots \\ LMP_{N_B} \end{bmatrix} = \begin{bmatrix} C_1 & \dots & C_{N_{MG}} \end{bmatrix} \times \mathbf{W_{MG}} \tag{7}$$

From (7), any combination of congestion pattern and marginal unit pattern (i.e., potential marginal patterns) produces LMPs. However, some obtained LMPs are invalid, which means some combinations are invalid. For a specific system, the number of units and the number of transmission lines are both limited, and thus, the number of potential marginal patterns and LMPs are also limited or finite. The enumeration of the potential marginal patterns gives all the possible LMP values. The number of combinations is given in (8).

$$N_{co} = \sum_{i=0}^{N_G} \frac{N_G!}{(N_G - i)!} \times \sum_{i=0}^{N_L} \frac{N_L!}{(N_L - i)!} \tag{8}$$

However, problem (1) is solvable only if the first matrix in (2) is invertible, which means the number of binding line flow constraints must equal the number of marginal units minus 1, as shown in (9). Therefore, the value of $N_{co}$ can be reduced as in (10).

$$UL + MG = 1 + CL + UL \tag{9}$$

$$N_{co} = \sum_{i=0}^{N_G} \left( \frac{N_G!}{(N_G - i)!} \times \sum_{j=0}^{i-1} \frac{N_L!}{(N_L - j)!} \right) \tag{10}$$

The above steps give all the potential values of LMPs by enumerating all the potential marginal patterns. However, some patterns are nonexistent under any load.

Problem (1) can be equivalently represented by a system of constrained equations (i.e., Karush-Kuhn-Tucker (KKT) conditions) [22]. Traditionally, the load $d_i$ is known, and solving the KKT system provides the value of the Lagrangian multipliers, which construct the values of LMPs. However, for given marginal patterns and LMPs, there may be multiple suitable load patterns. If any load pattern leads to such marginal patterns and LMPs, the obtained marginal pattern and LMPs are valid. Therefore, the load $d_i$ at each bus is treated as a variable to examine if there is a solution for (11) and (12).

$$\mathbf{LMP} = \lambda + \mathbf{GSF} \times \mu \tag{11}$$

$$KKT\ system\ of\ problem\ (1) \tag{12}$$

The KKT system is a necessary and sufficient condition for the convex problem (1). Therefore, if the KKT system is solvable, then the variables construct the optimal solution for the problem (1). In (11), the value of LMPs is specified. If any solution for (12) exists, there are corresponding LMPs and a marginal pattern for the solution. Thus, for each combination

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/OAJPE.2022.3149308, IEEE Open Access Journal of Power and Energy



from (10), equations (7), (11), and (12) are solved to remove invalid marginal patterns. Although the potential combinations are generally a large set, the possible number of congested lines is generally less than the number of branches, which could further reduce the value of $N_{co}$. For example, the ISO New England system has 2771 branches but the average active transmission constraint in January 2020, their winter peak month, only has 142 branches [11].

The detailed procedures of this comprehensive enumeration are shown in *Algorithm CE*, where CE stands for "comprehensive enumeration."

| **Algorithm CE** | Function CE (market model parameters, potential line of congestions) |
|---|---|
| **Input** | Market model parameters and potential line of congestions |
| **Output** | Marginal pattern and LMP library |
| 1 | Construct the set for all potential combinations of congestion patterns and marginal unit patterns with (10). |
| 2 | **For** each combination **do** |
| 3 | Obtain the potential LMPs with (7). |
| 4 | Solving equation set (11) and (12) |
| 6 | **If** (11) and (12) are solvable **do** |
| 7 | Record the marginal pattern and LMPs. |
| 8 | **Else** |
| 9 | Continue. |
| 10 | **End if** |
| 11 | **End for** |
| 12 | **Return** the marginal pattern and LMPs library |

A test system based on the PJM 5-bus system in [24]-[25] is provided to demonstrate the proposed *Algorithm CE*. The marginal pattern library for this test system is constructed by *Algorithm CE*, as shown in Table II (on the previous page). Any load sample will correspond to one of the marginal patterns in Table II. Future research will validate the implementation of *Algorithm CE* by comparing the results with Table II. Algorithm 1 provides a comprehensive enumeration method for collecting marginal patterns and LMPs. This test system contains 85 potential combinations, and 70 of them are invalid and removed. For example, unit 1, unit 4, and unit 5 cannot be marginal units simultaneously under any load pattern. The whole process takes 89.81s.

*Algorithm IS: an iterative search method*

*Algorithm CE* enumerates all the potential marginal patterns and then removes invalid patterns. When the system becomes larger, the number of potential marginal patterns becomes astronomical, making the validation process computationally expensive. Therefore, *Algorithm IS* aims to link one valid marginal pattern to another. Then, the marginal patterns and LMPs can be collected iteratively.

Equation (4) shows the incremental change in unit output and power flow with respect to incremental change in loads. When the incremental change in unit output and power flow are equal to the distance between the current value and the constraint limit (i.e., become binding), the required load increase at each bus can be represented as a matrix $\Delta\mathbf{d}$ as shown in (13). Each element in matrix **Dis** indicates a constraint that is one binding constraint away (denoted as "surrounding" marginal patterns) from the current marginal pattern. If a new binding constraint is identified, a new marginal pattern is found. Thus, if the load changes as indicated in the matrix's $\Delta\mathbf{d}$ column, all surrounding marginal patterns are obtained iteratively.

$$\begin{bmatrix}\Delta P_{MG}\\ \Delta S_{UL}\end{bmatrix} = \mathbf{Dis} = \mathbf{W}\times\Delta\mathbf{d} \quad \text{where}$$

$$\Delta\mathbf{d} = \begin{bmatrix}\Delta\mathbf{d}_{1,1} & \cdots\\ \cdots & \Delta\mathbf{d}_{N_b,UL+MG}\end{bmatrix} \text{ and } \mathbf{Dis} = \begin{bmatrix}P^{\max} - MP\\ L^{\max} - FL\\ L^{\min} - FL\end{bmatrix} \quad (13)$$

However, it should be noted that the sensitivity matrix **W** is only valid under the current marginal pattern, which means that although some columns in the matrix $\Delta\mathbf{d}$ may lead to a new marginal pattern, it does not correspond to the constraint as indicated in the matrix **Dis**. Under the assumption of the constant load participating factors, the value of the matrix $\Delta\mathbf{d}$ is deterministic. Thus, the constraint corresponding to a lower value of load increase is always reached first (i.e., the next binding constraint). Under the assumption of varying load participating factors, each element in the matrix $\Delta\mathbf{d}$ becomes a variable, and different participating factors correspond to different next binding constraints. Equation (13) is, however, always valid in terms of linking the current marginal pattern to other marginal patterns.

Thus, a bilevel optimization model can be constructed to determine if the surrounding marginal pattern is valid. As shown in problem (14), the upper level aims to find a valid load increase $\Delta\mathbf{d}$ under the current marginal pattern such that it can make the corresponding constraint in matrix **Dis** become binding, as shown in (14d)-(14f). The lower level is the original ED model in problem (1) with the load increase $\Delta\mathbf{d}$. Iteratively solving the optimization problem (14) with respect to the element in matrix **Dis** gives all the valid surrounding marginal patterns from the current marginal pattern. If the problem (14) is not solvable, then the obtained marginal pattern is not valid.

*Upper-level problem:*

$$\min \sum_i \Delta d_{i,j} \quad (14a)$$

$$\Delta d_i \geq 0, \forall i \in N_b \quad (14b)$$

$$W_{i,j} \times \Delta d_{i,j} = Dis_j \quad (14c)$$

If the element in **Dis** corresponds to a capacity constraint:

$$P_i = P^{\max}{}_i, \forall i \in M \quad (14d)$$

If the element in **Dis** corresponds to a negative line limit:

$$\sum_{i=1}^{N_b} GSF_{l-i}(P_i - d_i - \Delta d_{i,j}) = L_l^{\min} \quad \forall l \in L^n \quad (14e)$$

If the element in **Dis** corresponds to a positive line limit:

$$\sum_{i=1}^{N_b} GSF_{l-i}(P_i - d_i - \Delta d_{i,j}) = L_l^{\max} \quad \forall l \in L^p \quad (14f)$$

*Lower-level problem:*

$$problem\ (1)\ \text{with}\ d_i + \Delta d_{i,j} \quad (14g)$$

The obtained surrounding marginal patterns are recorded in the library. Next, one of the surrounding marginal patterns is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/OAJPE.2022.3149308, IEEE Open Access Journal of Power and Energy



selected to be the next step. To collect as many marginal patterns as possible, the closest marginal pattern, which is the one with the smallest loading level increase, is selected. Then, (13) is recalculated at the new marginal pattern. The search is performed iteratively until problem (14) is unsolvable for all elements in matrix **Dis** or all elements in matrix **Dis** have been stepped. Then, a new step is selected from the library, until all the patterns in the library have also been stepped. This algorithm searches around the current marginal pattern and collects marginal patterns iteratively, which may miss some marginal patterns during the search. Therefore, *Algorithm IS* could be performed iteratively under different initial marginal patterns until the library is sufficient. The detailed procedures of this collection method are shown in *Algorithm IS*, where IS stands for "iterative search."

| **Algorithm IS** | Function IS (market model parameters, initial marginal pattern) |
|---|---|
| **Input** | Market model parameters and initial load pattern |
| **Output** | Marginal pattern and LMP library |
| 1 | Include the initial marginal pattern in the library |
| 2 | **For** unvisited marginal pattern in the library d**o** |
| 3 | **While** true **do** |
| 4 | Solving (13) under current marginal pattern |
| 5 | **If** all elements in matrix **Dis** have been stepped **do** |
| 6 | Break |
| 7 | **End if** |
| 8 | **For** each element in matrix **Dis do** |
| 9 | Solving optimization problem (14) |
| 10 | Record marginal pattern and LMPs in the library |
| 11 | **End for** |
| 12 | Identify the least load increase |
| 13 | Step to the identified marginal pattern and denote it as stepped |
| 14 | **End while** |
| 15 | **End for** |
| **16** | **Return** the marginal pattern and LMPs library |

A test case is applied to demonstrate *Algorithm IS* via the European transmission network 89-bus system [26]. First, *Algorithm CE* is performed on this system to obtain all the marginal patterns, which indicates how many percentages of marginal patterns *Algorithm IS* can capture. The flow limits of three lines (line 34, line 66, and line 73) are considered. The remaining flow constraints are removed because excessive line limits prevent the implementation of *Algorithm CE*. The collected marginal patterns for *Algorithm CE*, *Algorithm IS*, and randomly generated by a uniform distribution are shown in Table III. The computation times for *Algorithm CE* and *Algorithm IS* are 133,237 s and 31,352 s, respectively. *Algorithm IS* collects 79% of the marginal patterns, with only 17% of the computation time of *Algorithm CE*. The randomly generated load sample has 5 million samples. By contrast, *Algorithm CE* only collects 12% of the marginal patterns at 274% of the computation time of *Algorithm IS*. The computation time of *Algorithm IS* is reduced significantly compared to *Algorithm CE*, and most marginal patterns are collected.

Table III. Comparisons of Algorithm CE and Algorithm IS

| | Algorithm CE | Algorithm IS | Random |
|---|---|---|---|
| Marginal Patterns | 841 | 663 | 105 |
| Computation Time | 133237s | 31352s | 85981s |

*Algorithm FS: a fast screening method*

The training dataset is usually generated offline, which may make the computation time of dataset generation a minor concern. However, a fast screening method is preferred for collecting marginal patterns when the system operates in complicated conditions. For example, solving the bilevel model in *Algorithm IS* becomes computationally expensive when the potential line of congestion is a large set.

*Algorithm FS* proposes a fast screening method, which is a variant of *Algorithm IS*. The iterative searching procedure of *Algorithm FS* is similar to *Algorithm IS*, but instead of solving the bilevel model (14), *Algorithm FS* only solves the $\Delta\mathbf{d}$ at the most sensitive bus for each element in the matrix **Dis**, as shown in (15). Solving the load increase at the most sensitive bus provides the smallest load increase.

$$\begin{bmatrix} Dis \\ \cdots \end{bmatrix} = \begin{bmatrix} W_L & \cdots \\ \cdots & \cdots \end{bmatrix} \times \begin{bmatrix} \Delta d & \cdots \\ 0 & \cdots \end{bmatrix} \quad (15)$$

Largest

Non-zero

All surrounding marginal patterns are scanned by solving the linear equation (15), which is much faster than solving the bilevel optimization model (14). Both *Algorithm IS* and *Algorithm FS* miss a few marginal patterns during the iterative collection process. However, *Algorithm FS* is an "incomplete" local search, meaning that part of the surrounding marginal pattern will also be missed, while *Algorithm IS* is a "complete" local search, which can obtain all surrounding marginal patterns. It is worth noting that the missing marginal patterns under the current step could still be collected in later steps.

The detailed procedures of this collection method are shown in *Algorithm FS,* where FS stands for "fast screening."

| **Algorithm FS** | Function FS (market model parameters, initial marginal pattern) |
|---|---|
| **Input** | Market model parameters and initial marginal pattern |
| **Output** | Marginal pattern and LMP library |
| 1 | Include the initial marginal pattern in the library |
| 2 | **For** unvisited marginal patterns in the library **do** |
| 3 | **While** true **do** |
| 4 | **If** all elements in matrix **Dis** have been stepped **do** |
| 5 | Break |
| 6 | **End if** |
| 7 | **For** each element in matrix **Dis do** |
| 8 | Solving equation (15) |
| 9 | Record marginal pattern and LMPs in the library |
| 10 | **End for** |
| 11 | Identify the least load increase. |
| 12 | Step to the identified marginal pattern and demote it as stepped |
| 13 | **End while** |
| 14 | **End for** |
| 15 | **Return** the marginal pattern and LMPs library |

*Algorithm FS* is also performed on the 89-bus system for the benefit of comparison to *Algorithm CE* and *Algorithm IS*. *Algorithm FS* collects 465 marginal patterns with a 4088s computation time, as shown in Table IV.

If compared to *Algorithm IS*, *Algorithm FS* further reduces the computational time, although some marginal patterns may

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/OAJPE.2022.3149308, IEEE Open Access Journal of Power and Energy



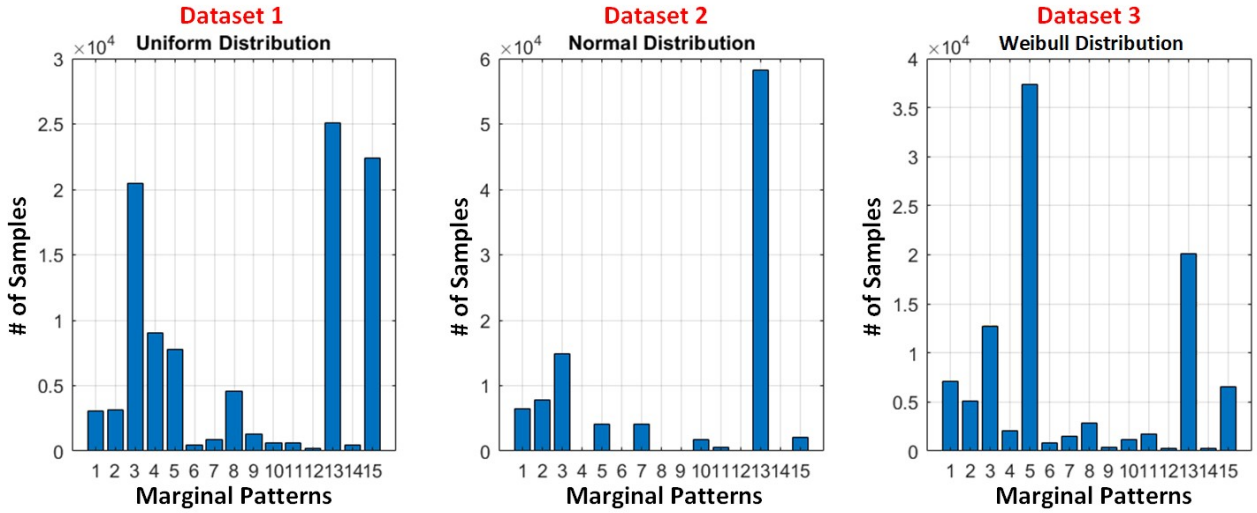


Fig. 2. Biased dataset based on pre-determined distributions

be missed. However, if compared with the random dataset generation, it collects more marginal patterns within a much shorter computational time. *Algorithm FS* is preferred if computational time of dataset generation is critical.

Table IV. Performance of Algorithm FS

| | Algorithm FS |
|---|---|
| Marginal Patterns | 465 |
| Computation Time | 4088s |

## B. Dataset Enhancement

Section A constructs the marginal pattern library using three different algorithms to examine the completeness of the training dataset. However, this is not the end of the effort. As previously mentioned, even if we identify many possible marginal patterns, the generated dataset may not contain samples in some CLL intervals, typically small ones. Thus, the samples can be biased.

Therefore, to fix the above potential problems, this section proposes *Algorithm DE*, where DE represents "dataset enhancement" with unbiased dataset generation.

The three algorithms (i.e., *Algorithms CE, IS,* and *FS*) in Section A construct a marginal pattern library. Marginal patterns in the unenhanced training dataset are compared with the marginal pattern library. For each marginal pattern in the library that does not exist in the training dataset, the following bilevel optimization model (16) is solved iteratively to generate extra samples to enhance the training dataset.

*Upper-level problem:*

$$\min \sum_i d_i + \beta_i \tag{16a}$$

$$\omega^{\min} \le \beta_i \le \omega^{\max} \tag{16b}$$

$$P_i = P^{\max}{}_i \times U_i, \forall i \in MG \tag{16c}$$

$$P^{\min}{}_i + \varepsilon \le P_i \le P^{\max}{}_i - \varepsilon, \forall i \in NG \tag{16d}$$

$$\sum_{i=1}^{N_b} GSF_{l-i}(P_i - d_i - \beta_i) = L_l^{\min} \quad \forall l \in L^n \tag{16e}$$

$$\sum_{i=1}^{N_b} GSF_{l-i}(P_i - d_i - \beta_i) = L_l^{\max} \quad \forall l \in L^p \tag{16f}$$

$$L_l^{\min} + \varepsilon \le \sum_{i=1}^{N_b} GSF_{l-i}(P_i - d_i - \beta_i) \le L_l^{\max} - \varepsilon \quad \forall l \in UL \tag{16g}$$

*Lower-level problem:*

$$problem\ (1)\ \text{with}\ d_i + \beta_i \tag{16h}$$

The goal of the upper-level problem (16a)-(16g) is to find a minimal loading level that leads to a specified marginal pattern. Constraint (16b) restricts the value of $\beta_i$, which results in different load samples at each iteration. In constraint (16c), the output of the non-marginal unit is restricted to either 0 or the maximum. In constraint (16d), the ε is a small positive value that restricts the generation of marginal units to be larger than 0 and smaller than the maximum. Similarly, the pattern of line flow is restricted through (16e)-(16g). Thus, this bilevel problem means that the upper-level problem tries to find a load sample, which makes the lower-level dispatch problem produce the marginal pattern as indicated in the upper-level constraints. Multiple load samples can be obtained by solving problem (16) iteratively. The obtained load samples are integrated to enhance the original training dataset. Then, the enhanced training dataset has enough load samples at all the marginal patterns. Thus, an unbiased dataset can be achieved to enhance the mapping library.

The detailed procedures of this enhancement method are shown in *Algorithm DE*. The effectiveness of *Algorithm DE* and the enhanced dataset are demonstrated with comparative case studies in Section V.

| **Algorithm DE** | Function DE (market model parameters, sample number ν) |
|---|---|
| **Input** | Market model parameters and sample number ν |
| **Output** | Samples to be included in the dataset |
| 1 | **For** each pattern that training dataset fails to include **Do** |
| 2 | $\omega^{\min}$ and $\omega^{\max} = 0$; |
| 3 | **While** ν! = 0 **do** |

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/OAJPE.2022.3149308, IEEE Open Access Journal of Power and Energy



| | |
|---|---|
| 4 | Solve the problem (16) |
| 5 | Record the load and the desired outputs |
| 6 | $\nu = \nu - 1$. |
| 7 | $\omega^{min} = \omega^{min} + \sigma$, $\omega^{max} = \omega^{max} + \rho \cdot \sigma$ |
| 8 | **End while** |
| 9 | **End for** |
| 10 | **Return** the recorded samples |

## V. Comparative Case Studies

The marginal pattern collection algorithms (i.e., *Algorithm CE*, *Algorithm IS*, and *Algorithm FS*) and the dataset enhancement algorithm (*Algorithm DE*) are discussed in the previous section. In this section, comparative case studies are presented to exemplify the biased dataset and demonstrate the superiority of the enhanced (unbiased) dataset obtained by *Algorithm DE*. Simulation runs were performed in MATLAB 2017 on a PC with an Intel i7-8650U processor and 16 GB RAM.

### *A. Insufficiency of the biased dataset*

Three different distributions including a uniform distribution, a normal distribution, and a Weibull distribution, are considered here for scenario sampling at each nodal load to demonstrate that the model-free approach based on randomized datasets is biased for ED-based problems. In *Algorithm CE*, all the marginal patterns for the modified PJM 5-bus system have been collected as in Table II.

One hundred thousand load samples are generated based on the above three distributions, and the corresponding marginal patterns are shown in Fig. 2. It should be noted that a few marginal patterns contain most of the samples in those datasets. For example, marginal pattern 13 contains more than half of the samples for the dataset generated by the normal distribution. However, most of the marginal patterns have insufficient samples. For example, marginal patterns 6, 10, 12, and 14 contain less than 400 training samples in any of the three datasets compared with the total of 100,000 samples.

Three neural networks (NN1, NN2, and NN3) with the same settings are trained with the above three different datasets, respectively. The load-LMP mapping is selected as a representative for the ED-based mapping problem. The neural networks are structured with two layers and 20 neurons under the Levenberg-Marquardt training algorithm. One hundred illustrative test samples are generated for marginal pattern 10 (small loading interval) and marginal pattern 13 (large loading interval), respectively.

Fig. 3 illustrates the prediction errors for patterns 10 and 13 in three different datasets. The x-axis sorts the test sample from the smallest error to the largest error. The average prediction errors for pattern 13 are 6.2% in NN1, 4.1% in NN2, and 6.1% in NN3, while the average prediction errors for pattern 10 are 29.2% in NN1, 34.1% in NN2, and 33.4% in NN3. This performance difference occurs because marginal pattern 10 contains considerably fewer samples than marginal pattern 13, as shown in Fig. 2. Thus, the prediction for marginal pattern 10 is much less accurate than for marginal pattern 13. Note, at this point, the *Algorithm DE* for dataset enhancement has not been applied. Fig. 3 shows poor performance due to a small number of training samples in a small loading interval (pattern 10) under biased dataset generation.

If the number of training samples in a marginal pattern is not enough, any type of input-output mapping in this marginal pattern will not be accurate because marginal patterns determine the optimal solution of ED-based problems. This insufficiency will be exacerbated in larger systems that have more marginal patterns. Note, in general, this phenomenon exists in any model-free application for ED-based problems due to the step change nature shown in Fig. 1, and this paper uses the neural networks for load-LMP mapping as an example.

The next subsection shows a comparison of the enhanced dataset (unbiased) and the randomly generated dataset (biased) in this subsection on the modified PJM 5-bus system and an 89-bus PEGASE system.

### *B. Comparison of the enhanced (unbiased) training dataset and biased training dataset*

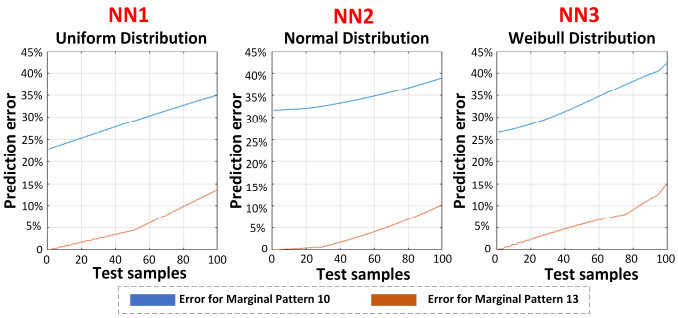


Fig. 3. Sorted prediction error for 100 test samples associated with patterns 10 and 13 by the three biased neural networks.

#### *1)* Modified PJM 5-bus system

A neural network is trained with the enhanced (unbiased) training dataset generated by *Algorithm DE*. In the enhanced dataset, 666 new samples are generated for each marginal pattern, which constitutes a total of 9,990 new samples. The same 100,000 initial training dataset samples (i.e., biased samples) as from the last subsection are applied, and the neural network trained by the enhanced dataset is compared with the neural network results trained by the biased dataset using uniform distribution-based sampling (i.e., NN1 in Fig. 3 in the previous subsection) as an example.

The predication errors for marginal pattern 10 (small loading interval) and marginal pattern 13 (large loading interval) are shown in Fig. 4. The x-axis sorts the test sample from the smallest error to the largest error. The prediction errors of the enhanced training dataset for marginal pattern 10 is 5.0% on average. In contrast, the prediction errors of NN1 in the previous subsection are 29.2% on average. Thus, the prediction error on marginal pattern 10 is significantly reduced because the enhanced dataset has filled more samples in marginal pattern 10, which has a narrow loading interval. The only cost is the new 9,990 training samples generated from *Algorithm DE*, which is less than 10% of the initial 100,000 samples in the biased training dataset, so *Algorithm DE* should be a helpful and worthy effort.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/OAJPE.2022.3149308, IEEE Open Access Journal of Power and Energy



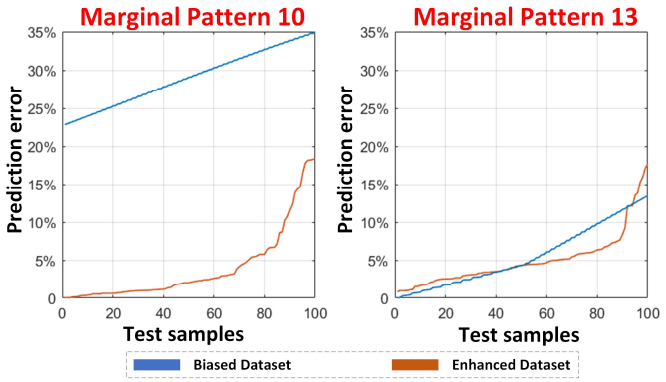


Fig. 4. Sorted prediction error comparison between the enhanced dataset and the biased dataset (from NN1) for the modified PJM 5-bus system.

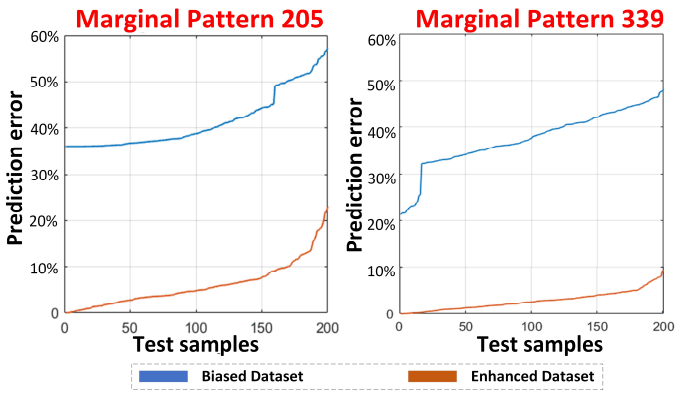


Fig. 5. Sorted prediction error comparison between the enhanced dataset and the biased dataset (from NN1) for the 89-bus systems.

However, the prediction accuracy on marginal pattern 13 on the enhanced, unbiased dataset is very close to the accuracy from the biased dataset. The errors are 3.9% vs. 5.5% on average, and 4.2% vs. 4.3% at median, respectively. The reason for this very minor improvement is that marginal pattern 13 has a large loading interval which already contains ample training samples in the biased dataset, and the extra dataset provided by *Algorithm DE* does not offer much help.

Note, although the comparison is carried out for NN1, similar conclusions hold for NN2 and NN3 since they have similar performance in prediction accuracy, as shown in Fig. 3.

*2)* 89-bus PEGASE system

Next, a similar prediction accuracy comparison between the enhanced (unbiased) training dataset and the biased training dataset is performed on the 89-bus PEGASE system. The biased dataset is the same as the example in Section IV, which contains 5 million samples. This dataset is enhanced with the marginal pattern library obtained in Section IV using *Algorithm DE*. Two marginal patterns (205 and 339) with small loading intervals are selected as illustrative examples, which are shown in Fig. 5. The x-axis sorts the test sample from the smallest error to the largest error. In the biased training dataset, patterns 205 and 339 contain less than 10 samples, and thus, the average prediction errors are 41.7% and 37.8%, which are extremely high. In contrast, the enhanced dataset adds 1500 extra training samples to each marginal pattern. This significantly reduces the average prediction error to 9.3% and 4.2%, respectively.

In both the 5-bus and the 89-bus systems, the enhanced (unbiased) dataset significantly improves the prediction accuracy for load-ED mapping in marginal patterns with small intervals (i.e., insufficient training samples).

Advanced learning techniques with the enhanced dataset will be investigated in future works, since the focus of this paper lies in the enhancement of dataset generation to provide unbiased training samples.

## VI. Conclusion

In this paper, we have identified a phenomenon that training datasets generated by pre-determined distributions are biased for load-ED mapping. Marginal patterns characterize the optimal solution of ED-based problems, and different marginal patterns differ significantly in size, which causes the dataset to overfill patterns with large sizes while underfilling patterns with small sizes. Thus, this paper proposes three marginal pattern collection algorithms to construct a marginal patterns library. Then, a dataset enhancement algorithm is proposed to generate unbiased samples for each marginal pattern in the library. The proposed algorithms and enhanced training dataset are illustrated and examined with the modified PJM 5-bus system and the 89-bus PEGASE system. The case studies clearly demonstrate the effectiveness of the proposed approach which significantly improves the prediction accuracy of data-driven load-ED mapping.

Our future work will combine the enhanced dataset with advanced learning techniques to provide a comprehensive model-free load-ED mapping platform.

## VII. Acknowledgement

The authors would like to acknowledge the support in part by the US Department of Energy CEDS Project “Watching Grid Infrastructure Stealthily Through Proxies (WISP)” under award number DE-OE0000899 and in part by the CURENT which is a US NSF/DOE Engineering Research Center funded by NSF award EEC-1041877.

The authors would also like to thank all WISP project team members for the useful discussions.

## VIII. References

[1] D. Silver et al., “Mastering the game of GO with deep neural networks and tree search,” *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[2] X. Kou *et al*., "Model-Based and Data-Driven HVAC Control Strategies for Residential Demand Response," in *IEEE Open Access Journal of Power and Energy*, vol. 8, pp. 186-197, 2021.

[3] F. Li and Y. Du, "From AlphaGo to Power System AI," *IEEE Power and Energy Magazine,* vol. 16, issue 2, pp. 76-84, Mar. 2018.

[4] L. Duchesne, E. Karangelos and L. Wehenkel, "Recent Developments in Machine Learning for Energy Systems Reliability Management," in *Proceedings of the IEEE*, vol. 108, no. 9, pp. 1656-1676, Sept. 2020.

[5] Y. Sun, X. Fan, Q. Huang, X. Li, R. Huang, T. Yin, and G. Lin, "Local Feature Sufficiency Exploration for Predicting Security-Constrained Generation Dispatch in Multi-area Power Systems," *17th IEEE*

*International Conference on Machine Learning and Applications (ICMLA)*, Orlando, FL, 2018.

[6] A. Zamzam and K. Baker, "Learning optimal solutions for extremely fast AC optimal power flow," 2019, *arXiv:1910.01213*. [Online]. Available: http://arxiv.org/abs/1910.01213.

[7] D. Owerko, F. Gama and A. Ribeiro, "Optimal Power Flow Using Graph Neural Networks," *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, 2020, pp. 5930-5934.

[8] F. Fioretto, T. W. K. Mak, and P. Van Hentenryck, "Predicting AC optimal power flows: Combining deep learning and lagrangian dual methods," no. 2, 2019, *arXiv:1909.10461*. [Online]. Available: http://arxiv.org/abs/1909.10461.

[9] X. Pan, T. Zhao, M. Chen and S. Zhang, "DeepOPF: A Deep Neural Network Approach for Security-Constrained DC Optimal Power Flow," *IEEE Transactions on Power Systems*, In-Press.

[10] N. Chiang and A. Grothey, "Solving security constrained optimal power flow problems by a structure exploiting interior point method," *Optimization and Engineering*, vol. 16, no. 1, pp. 49–71, 2015.

[11] Q. Zhang, F. Li, H. Cui, R. Bo and L. Ren, "Market-Level Defense Against FDIA and a New LMP-Disguising Attack Strategy in Real-Time Market Operations," in *IEEE Transactions on Power Systems*, vol. 36, no. 2, pp. 1419-1431, March 2021.

[12] A. Jahanbani Ardakani and F. Bouffard, "Prediction of Umbrella Constraints," *2018 Power Systems Computation Conference (PSCC)*, Dublin, 2018, pp. 1-7.

[13] D. Deka, S. Misra, "Learning for DC-OPF: Classifying active sets using neural nets," *2019 IEEE Milan PowerTech*, Milan, Italy, 2019, pp. 1-6.

[14] Y. Ng, S. Misra, L. A. Roald and S. Backhaus, "Statistical Learning for DC Optimal Power Flow," *2018 Power Systems Computation Conference (PSCC)*, Dublin, 2018, pp. 1-7.

[15] H. Wang, C. E. Murillo-Sanchez, R. D. Zimmerman and R. J. Thomas, "On Computational Issues of Market-Based Optimal Power Flow," *IEEE Trans. on Power Systems*, vol. 22, no. 3, pp. 1185-1193, Aug. 2007.

[16] Q. Zhang, F. Li, Q. Shi, K. Tomsovic, J. Sun and L. Ren, "Profit-Oriented False Data Injection on Electricity Market: Reviews, Analyses, and Insights," in *IEEE Transactions on Industrial Informatics*, vol. 17, no. 9, pp. 5876-5886, Sept. 2021.

[17] F. Li and R. Bo, "Congestion and Price Prediction Under Load Variation," *IEEE Trans. on Power Systems*, vol. 24, no. 2, pp. 911-922, May 2009

[18] X. Geng and L. Xie, "Learning the LMP-Load Coupling from Data: A Support Vector Machine Based Approach," in *IEEE Transactions on Power Systems*, vol. 32, no. 2, pp. 1127-1138, March 2017.

[19] Q. Zhou, L. Tesfatsion and C. Liu, "Short-Term Congestion Forecasting in Wholesale Power Markets," in *IEEE Transactions on Power Systems*, vol. 26, no. 4, pp. 2185-2196, Nov. 2011.

[20] Q. Zhang and F. Li, "Cyber-Vulnerability Analysis for Real-Time Power Market Operation," in *IEEE Transactions on Smart Grid*, vol. 12, no. 4, pp. 3527-3537, July 2021.

[21] R. Bo and F. Li, "Efficient Estimation of Critical Load Levels Using Variable Substitution Method," *IEEE Transactions on Power Systems,* vol. 26, no. 4, pp. 2472-2482, Nov. 2011.

[22] F. Ding, Y. Zhang, J. Simpson, A. Bernstein and S. Vadari, "Optimal Energy Dispatch of Distributed PVs for the Next Generation of Distribution Management Systems," in *IEEE Open Access Journal of Power and Energy*, vol. 7, pp. 287-295, 2020.

[23] A. J. Conejo and C. Ruiz, "Complementarity, Not Optimization, is the Language of Markets," in *IEEE Open Access Journal of Power and Energy*, vol. 7, pp. 344-353, 2020.

[24] F. Li and R. Bo, "DCOPF-based LMP Simulation: Algorithm, Comparison with ACOPF, and Sensitivity," *IEEE Transactions on Power Systems*, vol. 22, no. 4, pp. 1475-1485, Nov. 2007.

[25] F. Li and R. Bo, "Small Test Systems for Power System Economic Studies," *IEEE PES GM 2010,* Minneapolis, MN, Jul. 25-29, 2010.

[26] R. D. Zimmerman, C. E. Murillo-Sanchez, and R. J. Thomas, "Matpower: SteadyState Operations, Planning and Analysis Tools for Power Systems Research and Education," *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 12–19, Feb. 2011.

## IX. BIOGRAPHIES

**Qiwei Zhang** (S'17) is presently a Ph.D. student in the department of electrical engineering and computer science at The University of Tennessee, Knoxville, TN, USA. He received his B.S.E.E. degree from North China Electrical Power University in 2016 and M.S.E.E degree from UTK in 2018. His research interests include cyber security in power systems, power system optimization, and market operation.

**Fangxing Li** (S'98–M'01–SM'05–F'17) is also known as Fran Li. He received the B.S.E.E. and M.S.E.E. degrees from Southeast University, Nanjing, China, in 1994 and 1997, respectively, and the Ph.D. degree from Virginia Tech, Blacksburg, VA, USA, in 2001. Currently, he is the James W. McConnell Professor in electrical engineering and the Campus Director of CURENT at the University of Tennessee, Knoxville, TN, USA. His current research interests include renewable energy integration, demand response, distributed generation and microgrid, energy markets, and power system computing. Prof. Li is presently serving as the Editor-In-Chief of the *IEEE Open Access Journal of Power and Energy (OAJPE)* and the Chair of the IEEE/PES Power System Operation, Planning and Economics (PSOPE) Committee.

**Wei Feng** (S'12) received the B.S. and M.S. degrees from Tsinghua University, Beijing, China, and the University of Chinese Academy of Sciences, Beijing, China, in 2012 and 2015, respectively. He received the Ph.D. degree from University of Tennessee, Knoxville, TN, USA, in 2020. His research interests include high-performance computing, advanced analysis and optimal design in power systems.

**Xiaofei Wang** (S'20) received the B.S. degree from North China Electric Power University in 2014, and the M.S. degree from Wuhan University, China, in 2017. He is pursuing the Ph.D. degree at the University of Tennessee, Knoxville, TN, USA. His research interests include power system optimization, demand response, and distribution markets.

**Linquan Bai** (SM'20) received the B.S. and M.S. degrees in electrical engineering from Tianjin University, Tianjin, China, in 2010 and 2013, respectively, and the Ph.D. degree in electrical engineering from the University of Tennessee, Knoxville, TN, USA, in 2017. He is currently an Assistant Professor with the Systems Engineering and Engineering Management Department, University of North Carolina at Charlotte, Charlotte, NC, USA. His research interests include power system optimization and economics, distributed energy resources and integrated energy systems. He is an Associated Editor of the Journal of Modern Power Systems and Clean Energy.

**Rui Bo** (SM'10) received the BSEE and MSEE degrees in electric power engineering from Southeast University (China) in 2000 and 2003, respectively, and received the Ph.D. degree from the University of Tennessee, Knoxville, TN, USA in 2009. He is currently an assistant professor of the Electrical and Computer Engineering Department at Missouri University of Science and Technology (formerly University of Missouri-Rolla). He worked as a principal engineer and project manager at Mid-continent Independent System Operator (MISO) from 2009 to 2017. His research interests include computation, optimization and economics in power system operation and planning; high performance computing; electricity market simulation, evaluation and design.