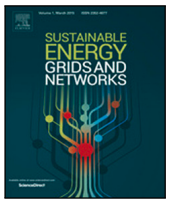
Sustainable Energy, Grids and Networks 42 (2025) 101672

Contents lists available at ScienceDirect

Sustainable Energy, Grids and Networks

journal homepage: www.elsevier.com/locate/segan



Machine learning-driven multi-agent-based AC optimal power flow with effective dataset creation for data privacy and interoperability

Burak Dindar [a], Can Berk Saner [b,*], Hüseyin K. Çakmak [a], Veit Hagenmeyer [a]

[a] Institute for Automation and Applied Informatics, Karlsruhe Institute of Technology, Karlsruhe, 76131, Baden-Württemberg, Germany
[b] Department of Mathematics, National University of Singapore, Singapore, 119076, Singapore

ARTICLE INFO

Keywords:
Data privacy
Effective dataset generation
Machine learning
Neural network
Optimal power flow

ABSTRACT

As power systems continue to evolve, the demand for effective collaboration among institutions has grown, driven by the challenges of balancing production and consumption, as well as by the increasing need for redispatch. Despite this, achieving interoperability in such a complex landscape is often hindered by concerns regarding data privacy. In response to these challenges, our paper presents a novel approach: a multi-agent system (MAS)-based AC optimal power flow (AC-OPF), empowered by machine learning (ML), designed for safeguarding data privacy and promoting interoperability. In the proposed method, the technical operator agent creates an effective dataset using n-ball, multivariate Gaussian distribution (MGD), and perturbation techniques. It also formulates valid inequalities to reduce the search space. Then, neural network (NN) models are developed to map the feasible space of the AC-OPF by utilizing only active power. Notably, these models conceal both the grid topology and sensitive data before transmission to another agent. Subsequently, the market operator agent integrates these NN models and valid inequalities into a mixed-integer linear programming (MILP) problem. This resulting MILP can be solved with various market based objective functions and constraints considering the power system limits. Thus, if there are private market-based data, they are kept confidential without being shared with the other agent. In addition, mapping is performed using the effective dataset generation technique that ensures a balanced representation of feasible and infeasible data points around the boundary. Furthermore, this effective dataset contributes to achieving remarkable accuracy in NN models, even with a relatively small volume of data points. The results on 30-, 57-, and 162-bus benchmark models of PGLib-OPF demonstrate that the proposed method can be effectively conducted while concurrently enhancing data privacy, and thus interoperability among institutions.

1. Introduction

Over the years, the electricity grid has undergone significant transformations for various reasons, including the reduction in fossil fuel usage and the emergence of new technological developments as power electronics. However, the increasing uncertainties and complexities of the grid owing renewable energy resources have made it challenging to maintain a balance between electricity production and consumption. Consequently, there is a growing need for flexibility and redispatch in the system to ensure optimal operation. To manage the electricity market effectively and prevent energy bottlenecks, it is crucial to resolve AC optimal power flow (AC-OPF) and make dispatch decisions [1]. The use of the AC-OPF and dispatch mechanisms in the electricity market has the potential to generate significant economic gains. For example, in the United States, increasing market efficiency by 5% through AC-OPF can result in annual savings of up to 6 billion dollars [2]. Given the expected increase in uncertainty in the near future, the use of effective AC-OPF and dispatch mechanisms can lead to even greater economic gains.

In the transformed electricity grid, transmission system operators (TSOs), distribution system operators (DSOs), distributed energy resources (DERs) companies, and aggregators must work together. For instance, in Germany's Redispatch 2.0, all conventional plants and DERs with an installed capacity of 100 kW or more, as well as DSOs, are required to participate in redispatch [3]. Additionally, in the future Redispatch 3.0, private customers will be allowed to participate in the market via aggregators [4]. Therefore, AC-OPF and redispatch should now be realized with the participation of different stakeholders, not only by system operators, as in the past. This will increase the need for cooperation among the institutions. To this end, various institutions,

* Corresponding author.
E-mail addresses: burak.dindar@kit.edu (B. Dindar), sanerc@u.nus.edu (C.B. Saner), hueseyin.cakmak@kit.edu (H.K. Çakmak), veit.hagenmeyer@kit.edu (V. Hagenmeyer).

https://doi.org/10.1016/j.segan.2025.101672
Received 26 July 2024; Received in revised form 6 February 2025; Accepted 28 February 2025
Available online 10 March 2025
2352-4677/© 2025 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

including universities, TSOs, DSOs, and consulting firms, are working together on research and development projects, such as Digiplat and Enera, to develop and implement effective cooperation mechanisms for flexibility and redispatch decisions [5,6].

Data privacy is one of the most important factors that complicates cooperation between industry partners. For instance, in a project that plans to establish a common platform for the redispatch mechanism, a TSO needs to share sensitive data, such as grid topology, voltage, and current magnitudes, with other partners that perform market research. Therefore, there is a need for a mechanism that protects data privacy between the model owner and other parties. To protect data privacy, institutions have developed various methods such as sharing simplified "light models" of the grid [7]. Nevertheless, these light models may still contain sensitive data, and it remains unclear to what extent simplified models can accurately represent the underlying power system. In the present study, a framework that prevents the sharing of sensitive data between institutions, who own grid models and do not own grid models, is the main objective to increase cooperation by providing data privacy.

In the literature, to preserve data privacy, a distributed OPF approach is used. In this approach, the problem is divided into subproblems to prevent complete grid model sharing. However, the complex voltage and/or active and reactive power of tie-lines are shared between neighboring regions [8,9]. Despite being less sensitive, this data could still pose privacy risks due to potential re-identification issues [10]: a privacy-preserving distributed OPF algorithm is developed and various encryption techniques are used to prevent violations. In [11] an encryption step is added to differentially private projected subgradient algorithm to preserve data privacy. With the added encryption step, OPF's accuracy remains the same, but it affects computation negatively by increasing the number of iterations. A privacy-preserving quadratic optimization algorithm is introduced in [12] to address security-constrained OPF in the smart grid. Although the number of iterations remains relatively unchanged, this method is 2^4 times slower compared to the non-privacy-preserving approach. In [10] a differentially private distributed algorithm is proposed to preserve re-identification of shared data. This method ensures the feasibility of the solution while adhering to constraints. However, its computation time is notably long; for instance, the 118-bus and 189-bus systems require 82.160 and 86.908 min, respectively. A partially homomorphic encryption technique is proposed in [13]. With this method, the OPF can be solved with several iterations, and the optimality gap is very small. While the paper presents an innovative approach to data privacy through encryption, its implementation may pose significant challenges, as the method cannot be readily solved through commercial solvers and lacks flexibility in accommodating additional constraints beyond standard optimal power flow constraints. As can be seen, there are trade-offs between data privacy and efficiency of the AC-OPF in the literature. However, for an effective privacy-preserving OPF, the computational burden should be low, the model and solution performance, and the flexibility of the method should be high. Therefore, innovative methods are needed. The method proposed in the present study ensures that all these requirements are satisfied.

Instead of using distributed optimization methods, an alternative approach for achieving privacy-preserving OPF involves employing machine learning models. Specifically, the party with access to the system model can train a machine learning model to obtain an equivalent representation of the system model. This trained model can then be shared with other parties instead of the original model, which can be used within the OPF framework to seek optimal power dispatch without requiring sensitive data. Establishing such a framework is the primary objective of the present study. In recent years, numerous studies have explored the use of machine learning-based approaches in the context of OPF problems. In general, these models are employed for "end-to-end" prediction, wherein they are trained to output the optimal generation dispatch for a given input of load distribution [14,15]. These methods are generally used to reduce the computational burden of OPF. In [16], an integration of deep NN and Lagrangian duality is proposed, enabling OPF to be solved in milliseconds for systems with up to 3400 buses while accounting for system constraints. A physics-informed, data-driven OPF approach utilizing a stacked extreme learning machine is introduced in [17], which improves the algorithm efficiency by simplifying the time-consuming parameter tuning process. Additionally, [18] develops a deep NN for high-dimensional load-to-solution mapping to solve the OPF, resulting in a significant acceleration of computation time. However, because these end-to-end approaches do not account for data privacy and their ML models are challenging to adapt to varying constraints and objective functions, they are not directly applicable to privacy-preserving OPF.

In order to achieve privacy-preserving OPF using an ML-based representation of the power system model, two essential components are required. Firstly, the ML model should accurately represent the underlying system model to ensure that the OPF results obtained via the ML-based representation do not differ significantly from the results obtained using the actual system model. Secondly, an efficient integration of the trained ML-model within an optimization formulation is necessary to ensure that the problem can be solved to (near-) optimality without undue computational expense.

The accuracy of the ML-based representation is largely contingent upon the representational power of the employed ML model, as well as the quality of the training data [19,20]. To ensure effectiveness, datasets should feature a balanced ratio of feasible/infeasible data points, including high-quality samples around decision boundary. Consequently, there is considerable research emphasis on effective dataset generation techniques. Recent advancements in this domain are summarized in [21]. Despite considerable efforts, many existing methods only represent a small portion of the feasible space, impacting overall model accuracy. Addressing these challenges, a new split-based method proposed in [21] aims to generate diverse data samples covering the entire feasible space, however it does not adequately address class imbalance. Another approach, outlined in [22], offers an efficient dataset generation method leveraging hyperplanes to eliminate large portions of the input space and create a balanced dataset. However, the required data points to produce a balanced dataset are too many and ML model accuracies are poor. Moreover, these methodologies do not guarantee the acquisition of high-quality data, particularly around the decision boundary, thereby hindering the accurate representation of the power system model using ML models.

In light of the aforementioned considerations, the present work presents a novel privacy-preserving multi-agent system (MAS)-based framework for solving the AC-OPF problem with the help of ML. Within the designed framework, the technical operator agent (TOA) creates NN models and a set of valid inequalities to represent the feasible set of AC-OPF of the power system model and its operational limits only with active power. Unlike existing approaches in the literature, these NN models are trained using an effective dataset generated through n-ball, multivariate Gaussian distribution (MGD) and perturbation techniques. These techniques are employed to ensure the creation of a balanced dataset containing high-quality data points around the boundary. Then, the NN models and set of valid inequalities are integrated into a mixed-integer linear programming (MILP) formulation by the market operator agent (MOA). Moreover, the MOA can incorporate various market-based objective functions and constraints into these MILP formulations. The resulting MILP enables the computation of AC-OPF without exposing sensitive power system data. In this approach, the TOA is not required to disclose any sensitive grid-related data, while the MOA does not need to share any special cost data and constraint with the TSO, if any. As a result, data privacy is ensured for both parties.

The key contributions of the present paper are as follows:

- The use of an NN representation in combination with a set of valid inequalities to represent the feasible set of AC-OPF in the MILP formulation prevents the sharing of sensitive data.

- An effective dataset generation methodology, which uses only active power, enables the training of an NN with accurate representation capability of the feasible set of an AC-OPF, even with a relatively small dataset and shallow NN architecture.
- The incorporation of the trained NN into a MILP formulation, which can be efficiently solved using commercial solvers. The resulting MILP is highly flexible, allowing for modifications to the objective function of the AC-OPF and the integration of additional constraints, as needed.

The rest of the paper is organized as follows: In Section 2, we describe the structure of the proposed MAS. In Section 3, we present the proposed methodology. In Section 4, we introduce an effective dataset creation technique. Subsequently, in Section 5, we detail the structure of the NN models and their transformation to the MILP. Then, we benchmark the proposed method with the AC-OPF, which does not consider data privacy, to evaluate its effectiveness through different case studies in Section 6. Finally, we provide our conclusions in Section 7.

2. Structure of the proposed multi-agent system

In the present paper, a multi-agent system (MAS) is introduced to ensure data privacy among institutions. The primary agents within this framework include the technical operator agent (TOA) and market operator agent (MOA).

The TOA is equipped with a comprehensive grid model and is responsible for the secure operation of the grid. Consequently, the TOA must adhere to various technical constraints, including voltage thresholds, line and transformer loading limits, as well as the active and reactive power constraints of the generators. The TOA can accurately estimate the active and reactive power requirements of the demand load at each bus by using its access to the complete grid model and historical grid data.

Based on this information, the TOA can define the feasible operational space of the power system. However, due to sensitive data such as the power system's topology and customer load information, sharing this data raises concerns about data privacy. Therefore, to safeguard data privacy, the feasible space of the power system is defined using NN models. Additionally, the TOA generates valid inequalities to reduce the search space and enables the MOA to solve the optimization problem more accurately and efficiently. These valid inequalities and NN models can be shared with the MOA conveniently, preserving the sensitive data.

The MOA lacks a grid model, but has bid/cost data and market constraints. This agent leverages this information to minimize dispatch or redispatch costs and maintain a balance between demand and generation through optimization techniques.

The MOA must consider the limits of the power system when conducting market based optimization. In this regard, the NN models developed by the TOA are used. Leveraging the structure of NNs, the MOA can integrate these models as constraints into its own optimization problem. By integrating these constraints alongside its own objective functions and constraints, the MOA can effectively solve the optimization problem while respecting the limits of the power system. The schematic representation of the proposed method is shown in Fig. 1.

Fig. 1 illustrates the process flow: The TOA begins by constructing a dataset to train the NN models and formulating valid inequalities. It is important to note that generic cost functions are employed during the training of these models, and an efficient dataset generation technique is designed. Following this, the trained NN models and valid inequalities are transferred to the MOA. Subsequently, the MOA integrates these NN models and valid inequalities as constraints within its own optimization problem.

As a result of this approach, the AC-OPF problem can be effectively addressed by considering both the power system and market constraints. Notably, by transmitting power system data to the MOA in the form of NN models, it guarantees the privacy of sensitive data. Furthermore, there is no requirement to share special constraints and cost data, if any, belonging to the MOA with the TOA, thereby further supporting data privacy. In this proposed method, the roles of the TOA and MOA can be fulfilled by various institutions, including universities, TSOs, DSOs, and consulting firms. Detailed information regarding the proposed method will be provided in the subsequent sections.

3. Overview of the proposed methodology

To set the notation in this study, parameters are denoted by standard letters (e.g., a, A), and variables are represented using boldface letters ($\boldsymbol{a}$, $\boldsymbol{A}$), while sets are indicated by calligraphic letters, such as $\mathcal{A}$. Functions are expressed by $A(\cdot)$. Scalar and (column) vector variables/parameters are presented in lowercase, while matrices are referred to by uppercase letters. The nth element of a vector a is denoted as $a^{(n)}$, whereas $a^{(-n)}$ represents all elements of a vector except the nth element, and the nth row of a matrix A is denoted as $A^{(n,:)}$. Furthermore, $\odot$ is the Hadamard product, and $\oslash$ is the Hadamard division.

3.1. Formulation of the standard AC-OPF

The standard AC-OPF problem for a power system with n_b buses and n_g generators can be formulated in a compact form as follows:

$$\min_{\boldsymbol{v},\boldsymbol{\theta},\boldsymbol{p}_g,\boldsymbol{q}_g} \quad F(\cdot) \tag{1a}$$

$$\text{s.t.} \quad G_P(\boldsymbol{v},\boldsymbol{\theta};Y_{bus}) + p_d - C_g\boldsymbol{p}_g = 0, \tag{1b}$$

$$G_Q(\boldsymbol{v},\boldsymbol{\theta};Y_{bus}) + q_d - C_g\boldsymbol{q}_g = 0, \tag{1c}$$

$$G_S(\boldsymbol{v},\boldsymbol{\theta};Y_{bus}) \le \overline{s}_f, \tag{1d}$$

$$\underline{v} \le \boldsymbol{v} \le \overline{v},\ \underline{\theta} \le \boldsymbol{\theta} \le \overline{\theta}, \tag{1e}$$

$$\underline{p}_g \le \boldsymbol{p}_g \le \overline{p}_g,\ \underline{q}_g \le \boldsymbol{q}_g \le \overline{q}_g, \tag{1f}$$

where $\boldsymbol{v},\boldsymbol{\theta} \in \mathbb{R}^{n_b}$ are the vectors of bus voltage magnitude, voltage angle, $\boldsymbol{p}_g,\boldsymbol{q}_g \in \mathbb{R}^{n_g}$ are the vectors of active and reactive power generations, C_g is the $n_b \times n_g$ *generation connection matrix* such that the element (i,j) is one if generator j is located at bus i, and zero otherwise. p_d, $q_d \in \mathbb{R}^{n_b}$ are the bus active and reactive power demand vectors, respectively, and Y_{bus} is the bus admittance matrix. In (1a), $F(\cdot)$ is the objective function. (1b) and (1c) represent the active and reactive balance equations, in which $G_P(\cdot)$ and $G_Q(\cdot)$ are the relevant functions. In (1d), $G_S(\cdot)$ denotes the line apparent power flows, which is bounded by the line flow limit vector $\overline{s}_f$. (1e) and (1f) imposes the upper and lower bounds on the variables.

3.2. Formulation of the machine learning-driven multi-agent-based AC-OPF

In the present section, we formulate the proposed privacy-preserving AC-OPF approach through collaboration between the TOA and MOA. Standard AC-OPF constraints (1b)–(1f) define a non-convex feasible set in $\boldsymbol{v},\boldsymbol{\theta},\boldsymbol{p}_g,\boldsymbol{q}_g$-space $\mathcal{S}$. However, this formulation encompasses sensitive data, such as network topology. The goal is to find an alternative but equivalent representation of $\mathcal{S}$ without revealing sensitive data, thereby enabling the sharing of this model with the MOA.

We begin by noting that the objective function $F(\cdot)$ in (1) typically depends only on $\boldsymbol{p}_g$, while $\boldsymbol{v},\boldsymbol{\theta},\boldsymbol{q}_g$ are involved only in constraints. Given this, the AC-OPF constraints (1b) - (1f) can be represented as a feasible set $\mathcal{P} \subseteq \mathbb{R}^{n_g}$. The set $\mathcal{P}$ includes all $\boldsymbol{p}_g$ values for which there

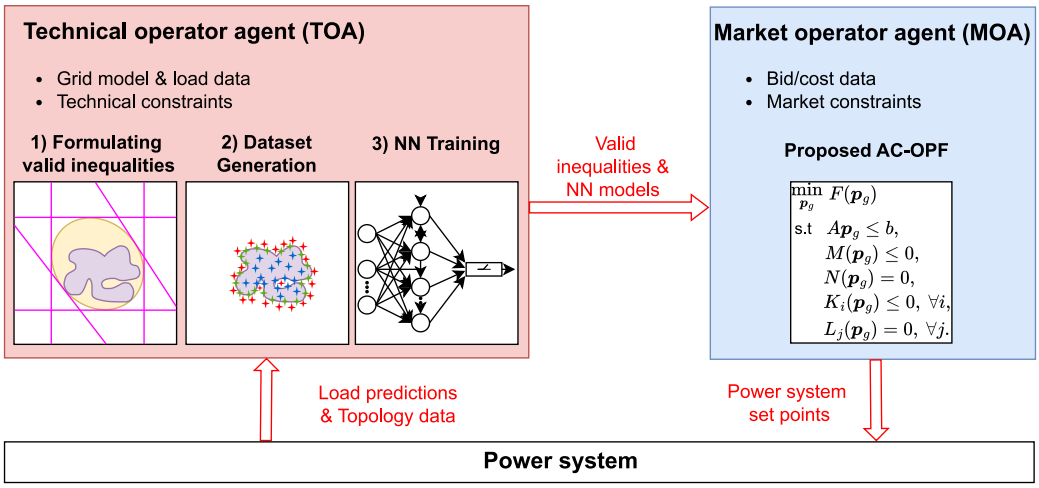


Fig. 1. The schematic representation of the proposed method.

exists at least one pair of (v, θ, q_g) that satisfies (1b) - (1f) [23,24]. Notably, $\mathcal{P}$ can be thought of as the projection of S onto the p_g-dimension. Then, (1) can be equivalently represented as:

$$\min_{p_g} \quad F(p_g) \tag{2a}$$

$$\text{s.t.} \quad p_g \in \mathcal{P}. \tag{2b}$$

For a general network, explicitly representing $\mathcal{P}$ mathematically is a complex task. In this work, our aim is to find an approximation of $\mathcal{P}$, denoted as $\hat{\mathcal{P}}$, which can be represented by a set of constraints that are functions of p_g. These functions are designed to avoid explicit parameterization by sensitive data exclusive to the TOA, allowing them to be shared with the MOA without data privacy concerns. Specifically, we hypothesize that $\hat{\mathcal{P}}$ takes the following form:

$$\hat{\mathcal{P}} = \left\{ p_g \mid Ap_g \le b,\ M(p_g) \le 0,\ N(p_g) = 0 \right\}, \tag{3}$$

where, $Ap_g \le b$ is a system of linear inequalities, which defines a convex polytope in $\mathbb{R}^{n_g}$ encompassing $\mathcal{P}$. The values of the matrix A and vector b are determined using *valid inequalities* derived from the convex relaxation of the AC-OPF, which is detailed in Section 4.1.

The function $M(\cdot)$ is designed to represent the feasibility condition of a generation dispatch p_g. In this regard, $M(\cdot)$ is constructed such that $M(p_g) \le 0$ holds only if $p_g \in \mathcal{P}$. To obtain $M(\cdot)$, we train a ML-based binary classifier model that classifies a given p_g as *feasible* (Class 0) if $p_g \in \mathcal{P}$, and *infeasible* (Class 1) otherwise.

Lastly, the function $N(\cdot)$ is aimed at capturing the power balance in the system. Specifically, $N(p_g) = 0$ holds only if the generation dispatch p_g meets the total load demand and system losses. To obtain $N(\cdot)$, we train a ML-based regressor model. This model takes $p_g^{(-s_g)}$ as the input, where s_g is the slack generator of the system, and outputs $p_g^{(s_g)}$. In particular, the regressor model can be represented as $N'(p_g^{(-s_g)}) = p_g^{(s_g)}$, therefore $N(p_g) = N'(p_g^{(-s_g)}) - p_g^{(s_g)}$.

To represent the functions $M(\cdot)$ and $N(\cdot)$, we propose using NN models for classification and regression, respectively. The reason of this choice is two-fold: on one hand, NNs have high generalization capabilities; on the other hand, the NN models can be exactly represented by a set of mixed-integer linear constraints [25], which enables us to integrate them into a mixed-integer linear programming (MILP) problems.

Indeed, the accuracy of approximation of $\hat{\mathcal{P}}$ for $\mathcal{P}$, particularly in terms of the NN-based functions $M(\cdot)$ and $N(\cdot)$, is contingent upon the quality of the training dataset. In Section 4, we introduce a rigorous workflow for creating the dataset consisting of highly informative samples. Using this workflow, within the proposed methodology, the TOA creates the dataset and trains the models for obtaining $M(\cdot)$ and $N(\cdot)$. As these functions are not explicitly parametrized by sensitive data, but rather by the weights and biases of the NN, they, as well as A and b, can be shared with the MOA without data privacy concerns. Additionally, even if an attempt is made to reconstruct the feasible space by reverse-engineering the NN models, it is important to note that these models are solely based on the active power outputs of the generators. As such, sensitive data, such as load demand, voltage profiles, and system topology, are not directly accessible through these models. Then, the MOA can solve the *machine learning-driven multi-agent-based AC-OPF problem* for the dispatch decisions, which is formulated as:

$$\min_{p_g} \quad F(p_g) \tag{4a}$$

$$\text{s.t.} \quad Ap_g \le b, \tag{4b}$$

$$M(p_g) \le 0, \tag{4c}$$

$$N(p_g) = 0, \tag{4d}$$

$$K_i(p_g) \le 0,\ \forall i, \tag{4e}$$

$$L_j(p_g) = 0,\ \forall j, \tag{4f}$$

where, (4b)–(4d) represent $\hat{\mathcal{P}}$ as given in (3). As indicated earlier, the NN-based functions in (4b) and (4c) can be represented as mixed-integer linear constraints, as detailed in Section 5. In (4e) and (4f), the functions $K_i(\cdot)$ and $L_j(\cdot)$ represent the additional inequality and equality constraints, respectively, defined by the MOA, if any. These constraints, for example, may stem from such as bilateral agreements, contracts, and market rules that the MOA wishes to keep confidential. Thanks to the proposed privacy-preserving formulation, the MOA does not need to disclose these constraints to the TOA and can incorporate them into the dispatch decisions. Furthermore, the MOA can freely define the objective function $F(p_g)$ without revealing it to the TOA, allowing for flexible and privacy-preserving adjustments based on bid or cost data. Moreover, the MOA can still reach the optimal solution of the freely defined objective function by leveraging the valid inequalities and NN models generated by the TOA, ensuring that the technical constraints are respected while maintaining privacy.

4. Valid inequalities and dataset creation

In the present section, our objective is to construct an effective dataset that enhances the performance of the NN models and elevates the accuracy of the proposed methodology. In the AC-OPF problem, the feasible space has a high-multidimensional non-convex structure and it is therefore quite complex to be analyzed [26,27]. In some networks, disconnected feasible spaces may even emerge [28].

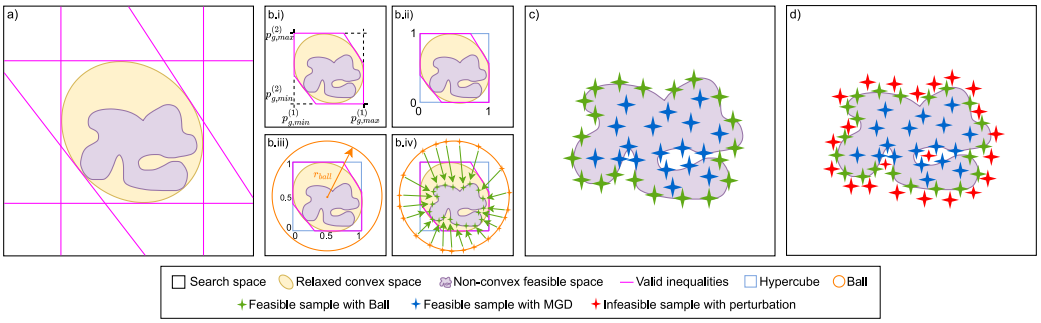


Fig. 2. The projection of the multi-step dataset creation methodology in two dimensions. a) Firstly, valid inequalities are established using SOCP relaxation, effectively delineating the area within which feasible data points reside. The region outside these valid inequalities is guaranteed to contain infeasible data points, effectively narrowing down the search space. Additionally, $p_{g,min}$ and $p_{g,max}$ are derived from these valid inequalities (see Algorithm 1). b.i) The relaxed convex space is restricted using $p_{g,min}$ and $p_{g,max}$. b.ii) All $\boldsymbol{p}_g$ values are normalized to fall within the range of 0 to 1, effectively confined within a hypercube. b.iii) A ball encompassing the hypercube is constructed, with a center located at 0.5 and a radius of r_{ball}. b.iv) Points are sampled from this ball, and through an optimization, the nearest feasible data points on the boundary of non-convex feasible space are determined (see Algorithm 2). c) Using MGD, feasible samples are acquired from the boundary, which cannot be sampled with the ball approach due to the complex non-convex nature of the feasible space. Additionally, data points within the feasible space are obtained through MGD (see Algorithm 3). d) Feasible data points are perturbed to generate infeasible data points around the boundary (see Algorithm 4).

Additionally, besides the non-convex structure of the AC-OPF, non-convexities also arise on the market side [29]. The inherent challenge is that data-driven approaches face significant barriers without a well-suited dataset. Methods developed without a suitable dataset generally map only a small portion of the feasible region, which causes the models to produce accurate results only in these regions [30]. However, for a good ML training process, it is essential to thoroughly map the feasible space and maintain a balanced dataset (feasible/infeasible) [21,31]. In particular, the errors made by the NN tend to increase near the boundaries, as expected [22]. Therefore, creating a dataset requires high-quality samples to map these boundaries appropriately [32].

Considering the aforementioned challenges, in the present paper we introduce a novel approach to dataset creation. As detailed in the previous section, the dataset is constructed using only the active power outputs of the generators $\boldsymbol{p}_g$. Our first step involves employing second-order cone programming (SOCP) relaxations to formulate valid inequalities, thereby reducing the search space (see Section 4.1). It is important to note that the SOCP relaxation is employed exclusively for generating valid inequalities, while all subsequent related calculations in this study are performed using the non-linear AC-OPF formulation. Subsequently, we advocate the utilization of the n-ball approach to generate feasible samples on the boundary of the $\mathcal{P}$ (see Section 4.2). Then, we apply multivariate Gaussian distribution (MGD) to the data points obtained in the previous section to generate samples within the $\mathcal{P}$, including undefined regions of the boundary (see Section 4.3). Finally, we introduce a perturbation approach to generate infeasible samples around the boundary by using feasible samples, generated in the previous steps (see Section 4.4). The projection of the multi-step dataset creation methodology is outlined in two dimensions in Fig. 2.

4.1. Generating valid inequalities

In the present section, we illustrate the methodology for formulating valid inequalities crucial for the optimization problem (4) solved by the MOA. It is worth noting that these valid inequalities are devised by the TOA in the form $A\boldsymbol{p}_g \leq b$. Indeed, the definition of space by valid inequalities plays a pivotal role in reducing the search space. In [22], the authors show that a significant portion of the search space in the AC-OPF problem comprises infeasible data points, which do not provide significant value. Instead, high-quality data points from the boundary between the feasible and infeasible regions should be prioritized. Therefore, narrowing down the search space using valid inequalities and focusing on the region bounded by these valid inequalities is crucial. This approach facilitates the attainment of quicker and more precise results in the optimization problem (4) addressed by the MOA. To achieve this, we employ the SOCP relaxation technique [33]. It is essential to note that the relaxed feasible space is used exclusively in the formulation of valid inequalities. The detailed process for generating valid inequalities is explained in Algorithm 1.

Algorithm 1 Valid inequalities generation

Input: Power system data
Output: A, b, $p_{g,min}$, $p_{g,max}$
Define an optimization problem according to SOCP relaxation

$$\min_{\boldsymbol{p}_g} \quad c^\top \boldsymbol{p}_g \qquad (5)$$
$$\text{s.t.} \quad \text{SOCP relaxation.}$$

$A \leftarrow [\,]; b \leftarrow [\,]$;
for c in $\{e_d, -e_d, o_d, -o_d\}$ **do**
  for $d \leftarrow 1$ to n_g **do**
    Solve (5) and obtain p_g^*
    $A \leftarrow [A; -c^\top]; b \leftarrow [b; -c^\top p_g^*]$;
  end for
end for
for c in $\{u, -u\}$ **do**
  Solve (5) and obtain p_g^*
  $A \leftarrow [A; -c^\top]; b \leftarrow [b; -c^\top p_g^*]$;
end for
$p_{g,min} \leftarrow -[b^{(1)} : b^{(n_g)}]$;
$p_{g,max} \leftarrow [b^{(n_g+1)} : b^{(2n_g)}]$;

For this, $A \in \mathbb{R}^{(4n_g+2)\times n_g}$ represents the matrix of coefficients, while $b \in \mathbb{R}^{4n_g+2}$ is the vector of constants. To construct the A matrix and b vector we formulate an optimization problem as shown in (5) where $c \in \mathbb{R}^{n_g}$ represents the coefficient vector. This optimization yields the optimal active power output of the generators, denoted as $p_g^* \in \mathbb{R}^{n_g}$ vector, based on the given c. We consider various linearly independent c vectors, namely e_d, o_d, and u. Here, $e_d \in \mathbb{R}^{n_g}$ denotes the standard unit vector, having a value of one at the $e_d^{(d)}$ element and zeros elsewhere; $o_d \in \mathbb{R}^{n_g}$ is a vector, with a value of zero at the $o_d^{(d)}$ element and ones elsewhere; and $u \in \mathbb{R}^{n_g}$ is the vector of ones.

The A matrix is constructed using $-c^\top$, and the b vector is created using $-c^\top p_g^*$ by leveraging the relaxation technique. When (5) is solved according to the SOCP relaxation, the optimal solution p_g^* must satisfy $c^\top p_g^* \leq c^\top \boldsymbol{p}_g$ for all feasible $\boldsymbol{p}_g$ due to the optimality condition of this convex problem. Therefore $-c^\top \boldsymbol{p}_g \leq -c^\top p_g^*$ is a valid inequality for the AC-OPF. It is important to emphasize that there cannot be even a single feasible sample outside the region defined by valid inequalities. In

addition, any sample that is infeasible according to relaxation remains infeasible for AC-OPF. Thereby, by employing valid inequalities, the search space is significantly reduced. Moreover, we use the first and second n_g elements of the b vector, corresponding to the e_d and $-e_d$ vectors, to derive $p_{g,min} \in \mathbb{R}^{n_g}$ and $p_{g,max} \in \mathbb{R}^{n_g}$ vectors, respectively, according to SOCP relaxation. These $p_{g,min}$ and $p_{g,max}$ vectors are then utilized to tighten the boundaries of $\boldsymbol{p}_g$. Subsequently, these refined boundaries are employed in the generation of the datasets.

4.2. N-ball approach

After establishing valid inequalities and reducing the search space, the next step is to obtain high-quality samples at the feasible/infeasible boundary of the AC-OPF. This is crucial for effectively defining the boundary and increasing the success of ML models. To achieve this objective, in this section, we focus on generating feasible data points at the boundary of the $\mathcal{P}$. We employ the n-ball approach for this purpose.

Our initial step involves normalizing each $\boldsymbol{p}_g$ using $p_{g,min}$ and $p_{g,max}$, such that $(\boldsymbol{p}_g - p_{g,min}) \oslash (p_{g,max} - p_{g,min})$. This normalization ensures that the $\mathcal{P}$ remains confined within a unit hypercube. Subsequently, we imagine a ball of dimension n_g that entirely encloses this unit hypercube. It is essential to note that for the ball to encompass the unit hypercube, its radius r_{ball} must satisfy $r_{ball} \geq \sqrt{2}/2$, and its center must be at 0.5. This configuration ensures that the ball covers the unit hypercube, thereby ensuring that the $\mathcal{P}$ remains inside the ball. Next, we project a sample taken from the surface of this n_g-dimensional ball onto the $\mathcal{P}$. This approach enables us to sample points from the boundary of the $\mathcal{P}$. To accomplish this, we formulate an optimization problem designed for identifying the closest feasible point on the AC-OPF to the sample taken from the ball. By iterative solving of this problem a specified number of times, we compile a dataset comprising feasible data points from the boundary. Algorithm 2 describes the creation of this dataset using the n-ball approach.

Algorithm 2 Feasible dataset generation with n-ball

Input: Power system data, $p_{g,min}$, $p_{g,max}$, n_{ball}, r_{ball}
Output: D_{ball}

Define an optimization problem according to standard AC-OPF

$$\min_{\boldsymbol{p}_g} \quad \left\| p_{g,ball} - \boldsymbol{p}_g \right\|_2^2 \tag{6}$$
$$\text{s.t.} \quad \text{(1b)} - \text{(1f)}.$$

$D_{ball} \leftarrow [\,]$;
for $idx \leftarrow 1$ to n_{ball} **do**
  $p_{g,ball} \leftarrow$ sample a vector of size n_g from $\mathcal{N}(0,1)$;
  $p_{g,ball} \leftarrow r_{ball} \times (p_{g,ball} / \left\| p_{g,ball} \right\|_2) + 0.5$;
  $p_{g,ball} \leftarrow p_{g,ball} \odot (p_{g,max} - p_{g,min}) + p_{g,min}$;
  Solve (6) and obtain p_g^*
  $D_{ball} \leftarrow [D_{ball}; p_g^{*\top}]$;
end for

For this, we start by generating a random vector $p_{g,ball} \in \mathbb{R}^{n_g}$ using a standard normal distribution. Given that we have normalized all $\boldsymbol{p}_g$, it is necessary to normalize $p_{g,ball}$ by dividing its L_2-norm. Afterwards, $p_{g,ball}$ needs to be scaled to represent the ball surface. For this purpose, we scale this vector by multiplying it by r_{ball}. Additionally, to ensure proper alignment with the center, we shift this vector by adding 0.5. This process yields a random point on the surface of the ball. Next, we denormalize this point using $p_{g,min}$ and $p_{g,max}$ to utilize it in the optimization problem. Finally, employing (6), we compute the feasible sample $p_g^* \in \mathbb{R}^{n_g}$ at the boundary of the AC-OPF that is closest to $p_{g,ball}$ with the assistance of the L_2-norm. By iteratively repeating this process n_{ball} times, we generate the dataset $D_{ball} \in \mathbb{R}^{n_{ball} \times n_g}$, which includes feasible samples from the boundary.

4.3. Multivariate Gaussian distribution

Following the sampling from the boundaries of the AC-OPF using the n-ball approach, our focus in this section is to sample from the boundaries that cannot be captured using the n-ball approach due to the intricate structure of the AC-OPF. Furthermore, we aim to obtain feasible samples from within the $\mathcal{P}$. To achieve this, we employ the multivariate Gaussian distribution (MGD). We create a dataset consisting of feasible samples using the MGD, leveraging the dataset generated with the n-ball approach and solving an optimization problem similar to the previous section. This process is explained in Algorithm 3.

Algorithm 3 Feasible dataset generation with MGD

Input: Power system data, D_{ball}, n_{mgd}
Output: D_f

Define an optimization problem according to standard AC-OPF

$$\min_{\boldsymbol{p}_g} \quad \left\| p_{g,mgd} - \boldsymbol{p}_g \right\|_2^2 \tag{7}$$
$$\text{s.t.} \quad \text{(1b)} - \text{(1f)}.$$

$\mu \leftarrow \text{mean}(D_{ball})$;
$\Sigma \leftarrow \text{cov}(D_{ball})$;
$D_{mgd} \leftarrow [\,]$;
for $idx \leftarrow 1$ to n_{mgd} **do**
  $p_{g,mgd} \leftarrow$ sample a vector of size n_g from $\mathcal{N}(\mu, \Sigma)$;
  Solve (7) and obtain p_g^*
  $D_{mgd} \leftarrow [D_{mgd}; p_g^{*\top}]$;
end for
$D_f \leftarrow [D_{ball}; D_{mgd}]$;

Utilizing the dataset D_{ball} generated in the previous section, we compute the row mean vector $\mu \in \mathbb{R}^{n_g}$ and the covariance matrix $\Sigma \in \mathbb{R}^{n_g \times n_g}$. Subsequently, a random vector $p_{g,mgd} \in \mathbb{R}^{n_g}$ is generated based on μ and Σ. By solving (7) with $p_{g,mgd}$ a feasible sample $p_g^* \in \mathbb{R}^{n_g}$ is determined. It is important to note that since this optimization is solved in accordance with the AC-OPF, the results are always feasible. This process is iterated n_{mgd} times to produce the dataset $D_{mgd} \in \mathbb{R}^{n_{mgd} \times n_g}$. Finally, by combining both datasets D_{ball} and D_{mgd}, we obtain a dataset $D_f \in \mathbb{R}^{n_f \times n_g}$ that effectively describes $\mathcal{P}$, comprising high-quality feasible data points. For this, n_f is defined as $n_f = n_{ball} + n_{mgd}$.

4.4. Perturbation approach

As the final step of dataset generation, we produce infeasible data points based on the feasible data points generated in the previous sections. In general, we introduce small perturbations to the feasible data points in the D_f dataset to derive infeasible data points. Given that the D_f primarily comprises feasible points from the boundaries, and these points undergo only minor perturbations, the resulting infeasible points also lie predominantly along the boundary. Consequently, we obtain both high-quality feasible and infeasible data points in the vicinity of the boundary, ensuring its accurate delineation. Additionally, it is worth mentioning that the NN regression model $N(\cdot)$ is employed to enhance the quality of the infeasible data points during the perturbation process. Further details regarding $N(\cdot)$ are provided in Section 5.2. This perturbation process and the production of infeasible dataset are summarized in Algorithm 4.

To generate an infeasible data point, we start by creating a random vector $p_{g,rand} \in \mathbb{R}^{n_g}$ through a standard normal distribution. Subsequently, we normalize $p_{g,rand}$ by dividing its L_2-norm, and determine its magnitude using $step \in \mathbb{R}$. Employing $p_{g,rand}$, we perturb the $p_{g,f} \in \mathbb{R}^{n_g}$ vector from the D_f, resulting the perturbed vector $p_{g,pert} \in \mathbb{R}^{n_g}$.

Algorithm 4 Infeasible dataset generation with perturbation

Input: Power system data, n_{ball}, n_{mgd}, D_f, $N(\cdot)$, lim, $step$
Output: D_{inf}, D

$D_{inf} \leftarrow [\,]$;
for $idx \leftarrow 1$ to $(n_{ball} + n_{mgd})$ **do**
  $p_{g,f} \leftarrow D_f^{(idx,:)^\top}$;
  $suc \leftarrow 1; ctr \leftarrow 0$;
  while $suc == 1$ and $ctr < lim$ **do**
    $p_{g,rand} \leftarrow$ sample a vector of size n_g from $\mathcal{N}(0,1)$;
    $p_{g,rand} \leftarrow step \times (p_{g,rand} / \left\| p_{g,rand} \right\|_2)$;
    $p_{g,pert} \leftarrow p_{g,f} + p_{g,rand}$;
    $p_{g,pert}^{(sg)} \leftarrow N'(p_{g,pert}^{(-sg)})$;
    if (1) with $p_g = p_{g,pert}$ is infeasible **then**
      $suc \leftarrow 0$
    end if
    $ctr \leftarrow ctr + 1$;
  end while
  if $suc == 0$ **then**
    $D_{inf} \leftarrow [D_{inf}; p_{g,pert}^\top]$;
  end if
end for
$D \leftarrow [D_f; D_{inf}]$;

However, rather than utilizing a completely random vector, we aim to use a vector that ensures the relationship between the slack generator and non-slack generators is maintained to increase the quality of the resulting perturbed vector. To achieve this, we leverage the NN regression model $N(\cdot)$. By predicting the slack generator $p_{g,pert}^{(sg)}$ using the non-slack generators $p_{g,pert}^{(-sg)}$, as determined by $N(\cdot)$, we update $p_{g,pert}$. Subsequently, by solving Eqs. (1) with the modified $p_{g,pert}$, we investigate its feasibility status. If the optimization yields an infeasible result, $p_{g,pert}$ is added to the infeasible dataset $D_{inf} \in \mathbb{R}^{n_{inf} \times n_g}$.

It is worth noting that $p_{g,f}$ is typically perturbed with a small magnitude represented by $step$ during the perturbation process. Consequently, $p_{g,pert}$ may always remain feasible. Thus, it is essential to limit the number of iterations to a specific value denoted as $lim \in \mathbb{Z}_+$. Following this iterative process, a balanced dataset $D \in \mathbb{R}^{(n_f + n_{inf}) \times n_g}$ is generated, comprising high-quality data points with a feasible-to-infeasible ratio of approximately one.

5. Neural network models and their transformation to MILP

We now discuss the training of the NN models for representing the functions $M(\cdot)$ and $N(\cdot)$, as well as the MILP representation of the trained NNs. We consider a feed-forward NN architecture with ReLU activation in the hidden layers. Without loss of generality, we focus on single hidden layer networks, but the concepts extend naturally to networks with multiple hidden layers.

The NN with an input $\boldsymbol{z} \in \mathbb{R}^{n_z}$ and an output $\boldsymbol{y} \in \mathbb{R}$ is represented as follows:

$$\boldsymbol{h} = \mathrm{ReLU}(W_h \boldsymbol{z} + \beta_h), \tag{8a}$$

$$\boldsymbol{o} = W_o \boldsymbol{h} + \beta_o, \tag{8b}$$

$$\boldsymbol{y} = \sigma(\boldsymbol{o}), \tag{8c}$$

where, $W_h \in \mathbb{R}^{n_h \times n_z}$ and $W_o \in \mathbb{R}^{1 \times n_h}$ are the weight matrices for the hidden and output layers, respectively, while n_h is the number of hidden nodes. The bias vectors are $\beta_h \in \mathbb{R}^{n_h}$ and $\beta_o \in \mathbb{R}$ for the hidden and output layers, respectively. The ReLU function is defined as $\mathrm{ReLU}(\cdot) = \max(0, \cdot)$. The activation function σ for the output layer is taken as the sigmoid function for the binary classification task, which is used to obtain $M(\cdot)$, whereas it is taken as a linear function for the regression problem, which is used to obtain $N'(\cdot)$, and therefore $N(\cdot)$.

5.1. Training of the classifier model

The NN binary classifier is trained to take a generation dispatch vector, $\boldsymbol{z} = \boldsymbol{p}_g$, as input and output zero if $\boldsymbol{p}_g$ is feasible, i.e., $\boldsymbol{p}_g \in \mathcal{P}$, and one otherwise. To train this network, we compile a training dataset consisting of feasible samples from D_f and infeasible samples from D_{inf}. Let the index set of the training dataset be denoted as $\mathcal{T}_M$. The loss function L_M, minimized during training, is a weighted binary cross-entropy loss defined as follows:

$$L_M = -\sum_{i \in \mathcal{T}_M} \omega_{10}\, y_i \log(\hat{y}_i) + \omega_{01}(1 - y_i) \log(1 - \hat{y}_i), \tag{9}$$

where $y_i \in \{0, 1\}$ is the true label (i.e., 0 for feasible, and 1 for infeasible instances) and $\hat{y}_i$ is the predicted label for instance $i \in \mathcal{T}_M$. The parameter $\omega_{10} > 0$ represents the weight assigned to incorrectly classifying an infeasible instance as feasible, while $\omega_{01} > 0$ represents the weight for the opposite error.

In this problem context, distinguishing the significance of incorrect predictions by the NN is crucial. Predicting a feasible point as infeasible incurs an economic loss, whereas predicting an infeasible point as feasible can lead to undesirable outcomes in the power grid. To mitigate the risk of identifying an infeasible point as feasible, we set the weights such that $\omega_{10} > \omega_{01}$. Thus, after all these steps, the function $M(\cdot)$ can be constructed.

5.2. Training of the regressor model

The NN regressor is trained to take a generation dispatch vector with the entry associated with the slack generator removed, $\boldsymbol{z} = \boldsymbol{p}_g^{(-s_g)}$, as input and output the power output of the slack generator, $\boldsymbol{p}_g^{(s_g)}$. To compile the training dataset for the regressor model, we consider only the feasible samples D_f. From these, we extract the rows corresponding to non-slack generators as the predictors and the slack generator as the target variable. Let the index set of the training dataset be denoted as $\mathcal{T}_N$. The mean squared error loss function L_N, minimized during training, is defined as follows:

$$L_N = \frac{1}{|\mathcal{T}_N|} \sum_{i \in \mathcal{T}_N} (\hat{y}_i - y_i)^2, \tag{10}$$

where y_i is the true value of the slack generator's power output and $\hat{y}_i$ is the predicted value for instance $i \in \mathcal{T}_N$. Finally, after the training process the function $N(\cdot)$ is obtained.

5.3. Neural network MILP representation

Once the NNs are trained and their weights and biases are identified, they are integrated into the optimization problem as constraints (4c) and (4d). To achieve this, we first employ the approach given in [25], which exploits the piecewise-linear nature of the ReLU function to obtain a mixed-integer linear representation. The set of constraints to represent (8a)–(8b) is given as follows:

$$W_h \boldsymbol{z} + \beta_h = \boldsymbol{h} - \boldsymbol{h}_-, \tag{11a}$$

$$0 \le \boldsymbol{h} \le \overline{h} \odot \boldsymbol{u}, \tag{11b}$$

$$0 \le \boldsymbol{h}_- \le \overline{h}_- \odot (1 - \boldsymbol{u}), \tag{11c}$$

$$\boldsymbol{o} = W_o \boldsymbol{h} + \beta_o, \tag{11d}$$

$$\underline{z} \le \boldsymbol{z} \le \overline{z}, \quad \underline{o} \le \boldsymbol{o} \le \overline{o}, \tag{11e}$$

$$\boldsymbol{h}, \boldsymbol{h}_- \in \mathbb{R}^{n_h}, \quad \boldsymbol{u} \in \mathbb{B}^{n_h}. \tag{11f}$$

The constraints (11a) to (11c) model the ReLU activation. Constraint (11a) decomposes the pre-activation output into the positive part $\boldsymbol{h}$ and the non-activated part $\boldsymbol{h}_-$. Constraints (11b) and (11c) ensure that $\boldsymbol{h}$ and $\boldsymbol{h}_-$ follow the ReLU behavior based on the binary

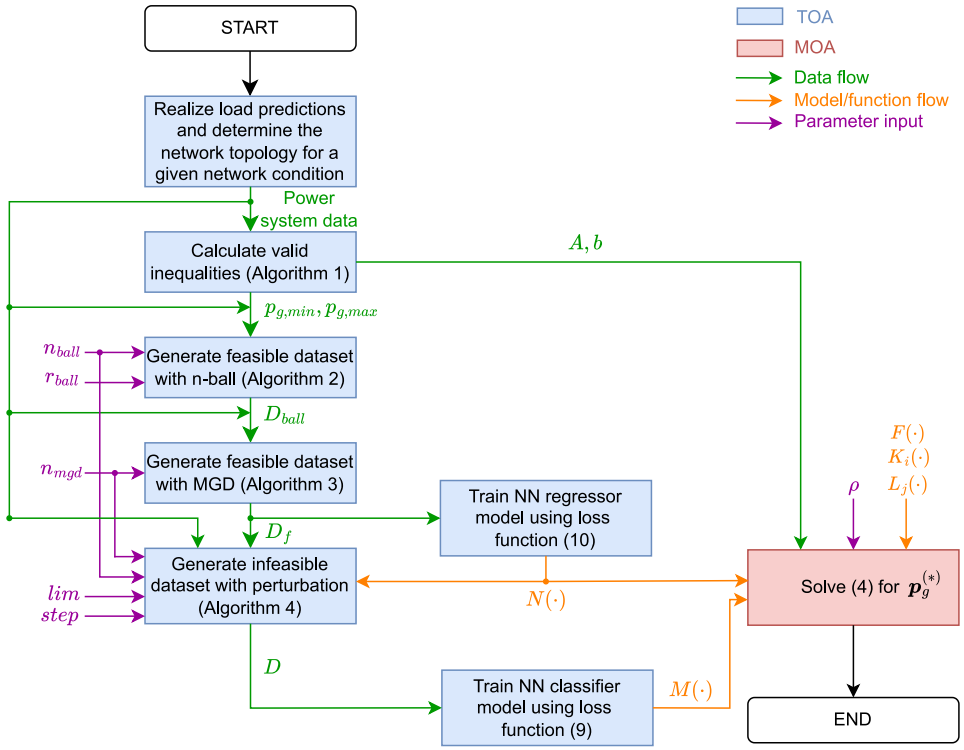


Fig. 3. Flowchart for the proposed methodology.

variables $\boldsymbol{u}$. Constraint (11d) defines the linear transformation of the output layer: Constraints (11e) ensure the variables are within specified bounds, and (11f) specifies the domains of the variables.

Then, to represent (4c), the following constraint is used together with (11):

$$\boldsymbol{o} \leq -\rho, \tag{12}$$

where, $\rho \geq 0$ represents a *conservativeness* parameter. In the context of the NN classifier, this parameter is used to ensure that the output $\boldsymbol{o}$ remains below a certain threshold, thereby providing a safety margin. This margin helps to account for inaccuracies in the NN predictions, ensuring that the generation dispatch remains within safe and feasible limits.

In order to represent (4d), the following constraint is used together with (11)[1]:

$$\boldsymbol{o} - \boldsymbol{p}_g^{(s_g)} = 0. \tag{13}$$

Finally, the flowchart of the entire process, as described in the preceding sections, is presented in Fig. 3.

6. Case studies and discussion

In our study, we evaluate the performance of our proposed method by comparing it to the AC-OPF, which does not consider data privacy, using the 30-, 57-, and 162-bus benchmark models from PGLib-OPF [34]. The power system data is used exactly as it is. Our evaluation process begins with the generation of an effective dataset, employing the techniques outlined in the previous sections to acquire high-quality data points on the boundary of the $\mathcal{P}$. Subsequently, we proceed to train NN models to map the $\mathcal{P}$, assessing the quality of their approximations. Finally, we test the efficacy of our proposed method by conducting comparative analyses against the AC-OPF, using various objective functions and constraints.

We select MATLAB as the primary simulation environment for our case studies. We utilize MATPOWER [35] with KNITRO solver [36] for the AC-OPF, while GUROBI [37] and YALMIP [38] are employed for the MILP. Furthermore, we train and test our NN models using TensorFlow/Keras [39,40]. The case studies are conducted on a PC equipped with an Intel Core i7-10700K CPU @ 3.80 GHz and 32 GB RAM.

6.1. Dataset creation process

We create datasets by following the steps explained in detail in Section 4. Firstly, we generate valid inequalities according to Algorithm 1. When examining the Algorithm 1, it can be observed that the dimension of matrix A is $4n_g+2\times n_g$ and vector b is $4n_g+2$. Therefore, for 30-, 57-, and 162-bus power systems, A has dimensions 26×6, 30×7, and 50×12, while b has dimensions 26, 30, and 50, respectively.

After generating valid inequalities, we produce feasible samples at the boundary of the $\mathcal{P}$ using the n-ball approach, following Algorithm 2. Specifically, we set $r_{ball} = 3$ and adjust n_{ball} for each case, as detailed in Table 1. Next, we apply the multivariate Gaussian distribution to D_{ball} as described in Algorithm 3, generating feasible data points from both the unsampled boundary and within the $\mathcal{P}$. It is worth noting that $n_{ball} = n_{mgd}$, so the size of the D_{mgd} and D_{ball} is equal. Subsequently, we obtain D_f by combining these two datasets containing feasible samples. Finally, we generate infeasible samples by perturbing the feasible samples according to Algorithm 4, creating D_{inf}. We select $lim = 5$ and $step = 5$ for this perturbation process. Consequently, D is obtained by combining D_f and D_{inf}. After all these steps, the generated number of data points and the feasibility ratio for each case are summarized in Table 1.

When examining Table 1, it can be observed that a balanced dataset of approximately 50% feasibility ratio is created for each case. The

[1] For brevity, the same set of variables is used for the NN models in this explanation. However, in practice, two distinct sets of variables are used: one for the classifier and another for the regressor, with the corresponding weights and biases.

Table 1
Summary of the dataset creation.

Power system	D_{ball}	D_{mgd}	D_f	D_{inf}	D	Feasibility ratio
Case 30	7500	7500	15 000	10 478	25 478	58.87%
Case 57	15 000	15 000	30 000	24 160	54 160	55.39%
Case 162	50 000	50 000	100 000	99 834	199 834	50.04%

Table 2
Metrics for the NN classification.

Power system	Accuracy	Recall	Specificity
Case 30	98.64%	98.45%	98.90%
Case 57	90.32%	84.40%	97.76%
Case 162	97.64%	96.17%	99.11%

Table 3
Metrics for the NN regression.

Power system	RMSE	MAE	MAPE
Case 30	0.529	0.373	0.176%
Case 57	0.686	0.334	0.424%
Case 162	0.227	0.119	0.025%

slight deviation from exactly 50% can be attributed to the fact that infeasible data points may not always be reachable during the perturbation process (see Algorithm 4). Therefore, the size of D_{inf} may be smaller than D_f. Additionally, the proposed data generation techniques facilitate the acquisition of high-quality samples around the boundary. Consequently, suitable datasets are generated for effective utilization in ML models with a relatively small number of data points.

6.2. ML training and approximation quality

After acquiring the datasets, we proceed to train the ML models to construct the functions $M(\cdot)$ and $N(\cdot)$. Subsequently, we assess the approximation quality of these models using test sets. Consistent with the typical procedure, we partition the datasets into 80% train set and 20% test set for both classification and regression models. As detailed in Section 5.1, we employ an NN classification model to classify feasible and infeasible samples. For each case, we utilize a single hidden layer and 150 hidden nodes. Additionally, we set w_{10} to 2.5, 5, and 10 for the respective power systems, while $w_{01} = 1$ for all systems.

Once we train the model, we evaluate the approximation quality of the NN classification model using accuracy, recall and specificity metrics [41] as shown in Table 2. As outlined in Section 5.1, we assign a higher weight value to predicting an infeasible point as feasible, aiming to prevent the model from misclassifying infeasible points. Consequently, the specificity metric, which indicates the prediction performance of infeasible points, is quite high. However, due to these weights, the accuracy and recall metrics are slightly lower. Although this situation may lead to some economic loss, it enhances the feasibility of the proposed method as we design it for.

The training process for the NN regression model follows a similar approach as explained in Section 5.2. We choose a single hidden layer with 500 hidden nodes for all power systems. It should be noted that only D_f is used for training the NN regression model. Subsequently, we evaluate the model's performance using numerical metrics such as root mean square error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE), as presented in Table 3. As demonstrated in the results, the NN regression models achieve performance that are close to 100% accuracy for each power system. Overall, the high performance achieved by both the NN classification and regression models indicates that the functions $M(\cdot)$ and $N(\cdot)$ collectively can effectively capture the behavior of the power systems using only the active power outputs of the generators $\boldsymbol{p}_g$.

Table 4
Comparison of feasibility ratio and average cost difference for different cost coefficients.

Power system	Feasibility ratio	Average cost difference
Case 30	100%	0.154%
Case 57	100%	0.006%
Case 162	96.40%	0.786%

6.3. Benchmark results with different cost coefficients

In this section, we assess the efficiency of the proposed method by conducting a comparative analysis with the AC-OPF, which does not consider data privacy, in terms of feasibility ratio and total cost. To achieve this, we randomly generate 1,000 different sets of cost coefficients for each power system. This approach allows for a detailed evaluation of the proposed method and represents various special objective functions that the market operator agent may have. We consider only linear cost coefficients as in PGLib-OPF library [34] without loss of generality. Here, we use conservativeness only for the most complex model, Case 162, as $\rho = 10$.

For each set of randomly generated cost coefficients, we first obtain a solution using the proposed method and subsequently validate its feasibility in the AC-OPF formulation. During the validation process, we apply a tolerance of 1×10^{-2} to avoid numerical errors. Then, we compare the total costs obtained from both methods for the same set of cost coefficients. The comparison results for feasibility ratio and average cost difference are presented in Table 4.

As evident from the results, the proposed method achieves a 100% feasibility ratio for Case 30 and 57, while achieving a slightly lower ratio of 96.40% for Case 162. Similarly, the average cost difference between the proposed method and the AC-OPF remains negligible, closely approaching 0% for Cases 30 and 57, and only reaching 0.786% for Case 162. For a more comprehensive analysis of the numerical results, the histogram of the total cost differences is depicted in Fig. 4.

Negative values in the histogram indicate instances where the proposed method achieves a lower total cost compared to the AC-OPF. For Case 30 and 57, the maximum cost difference is approximately 1%. Although Case 162 shows some high values, these instances are relatively rare, with only 10 out of 1,000 samples exceeding a 5% cost difference. This situation can be associated with the use of high weights and conservativeness to ensure a high feasibility ratio for this power system. Overall, the proposed method demonstrates performance that is comparable to the AC-OPF, while effectively preserving data privacy.

6.4. Benchmark results with different cost coefficients and constraints

In this section, similar to the previous one, we randomly generate 1,000 different sets of cost coefficients for each case study. However, in addition to this, we introduce different combinations of constraints. Consequently, we compare the proposed method with the AC-OPF, which does not consider data privacy, across various cost coefficients and constraints. This approach allows to represent special constraints along with the special objective functions that the market operator agent may have, providing a comprehensive evaluation of the method's performance. Rather than focusing on Case 30 and 57, we direct our attention to the more intricate Case 162 to provide a more comprehensive assessment of the proposed method's efficiency. To facilitate this investigation, we set the following three constraints:

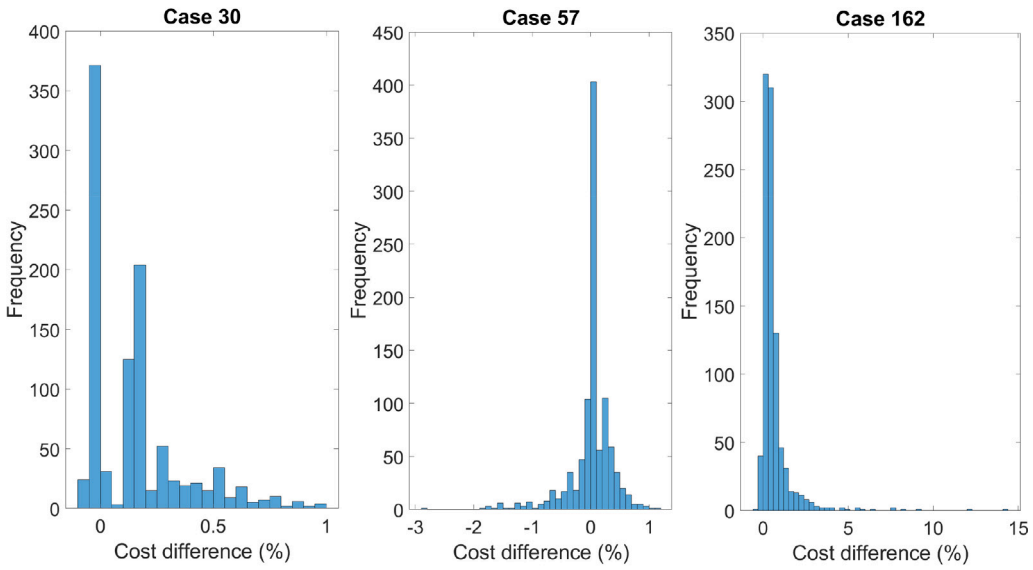


Fig. 4. The histogram of the total cost differences for different cost coefficients.

Table 5
Comparison of feasibility ratio and average cost difference for different cost coefficients and constraints.

Constraint combinations	Feasibility ratio	Average cost difference
(14a)	95.20%	0.790%
(14a) + (14b)	94.70%	0.739%
(14a) + (14b) + (14c)	95.70%	0.793%

$$p_g^{(11)} + p_g^{(12)} - 1000 \geq 0, \tag{14a}$$

$$p_g^{(7)} - p_g^{(4)} \geq 0, \tag{14b}$$

$$2 \times (p_g^{(2)} + p_g^{(8)}) - p_g^{(1)} \geq 0. \tag{14c}$$

In the standard case, the active power lower limits of the 11th and 12th generators, $\underline{p}_g^{(11)}$ and $\underline{p}_g^{(12)}$ are 0 MW. In (14a), we modify this by imposing a constraint that their total active power output must be greater than 1,000 MW. Additionally, although the upper limits of the active power for the 7th and 4th generators are $\overline{p}_g^{(7)} = 308$ MW and $\overline{p}_g^{(4)} = 366$ MW, we add a constraint that the $p_g^{(7)}$ must always be greater than $p_g^{(4)}$ in (14b). As a final constraint, we specify that twice $p_g^{(2)}$ plus $p_g^{(8)}$ must be greater than $p_g^{(1)}$ in (14c), while $\overline{p}_g^{(1)}, \overline{p}_g^{(2)}, \overline{p}_g^{(8)}$ are 1,147 MW, 451 MW, and 455 MW, respectively. The analyses are performed by adding these constraints incrementally, and the feasibility ratio and average cost difference, in comparison to the AC-OPF, are summarized in Table 5.

When the results are examined, it can be observed that, similar to the previous section, a feasibility ratio of around 95% is achieved in each case, and the average cost difference remains approximately 0.8%. This indicates that the proposed method maintains its effectiveness even with the introduction of various additional constraints. The histogram of the total cost differences is presented in Fig. 5.

Furthermore, when examining the histogram, it is evident that the distributions are similar across cases and even consistent with the previous section. This suggests that the inclusion of additional constraints does not impact the performance of the proposed method. Consequently, it can be inferred that various other constraints, beyond those specified, can be incorporated without any degradation in performance. In summary, the proposed method demonstrates acceptable performance compared to the AC-OPF while maintaining data privacy. Moreover, the integration of the trained NN model into the MILP formulation enables flexible utilization of different cost coefficients and additional constraints.

7. Conclusion

The need for interoperability in power system management is becoming increasingly critical to improve operational efficiency and reduce costs. For the effective use of mechanisms such as AC optimal power flow (AC-OPF) and redispatch, collaboration among institutions is key. However, achieving interoperability remains a challenge, primarily due to concerns surrounding data privacy. In the present paper, we introduce a multi-agent system (MAS)-based AC-OPF approach leveraging neural network (NN) models.

In the proposed method, an effective dataset is initially generated through the utilization of n-ball, multivariate Gaussian distribution, and perturbation techniques. Typically, the generation of a vast number of data points is required to procure high-quality data points from the feasible/infeasible boundary using traditional methods. However, the proposed dataset generation technique facilitates the creation of a balanced dataset, comprising high-quality data points around the boundary with relatively fewer data points. Leveraging this effective dataset, NN models that map feasible space using only active power, achieve high performance. By employing these NN models, sensitive data such as grid topology, current, voltage values, etc., are concealed, ensuring data privacy.

Subsequently, the NN models are integrated into a mixed-integer linear programming (MILP) formulation. Thanks to the NN structure, this integration enables the optimization problem to be addressed with varying market-based objective functions and constraints. Moreover, the incorporation of valid inequalities further enhances the efficiency of the optimization process by reducing the search space, leading to more accurate and expedited results. Consequently, the proposed approach enables the realization of the AC-OPF while effectively accounting for both market and system constraints.

The proposed method is evaluated using 30-, 57-, and 162-bus benchmark models from the PGLib-OPF. The test results reveal high performance in terms of NN accuracy and overall results, underscoring the efficacy of the proposed method. Consequently, the method enables the execution of AC-OPF while simultaneously ensuring data privacy for all stakeholders, thus fostering interoperability among institutions.

The potential of the proposed method can be further increased through the utilization of high-performance computing and advanced

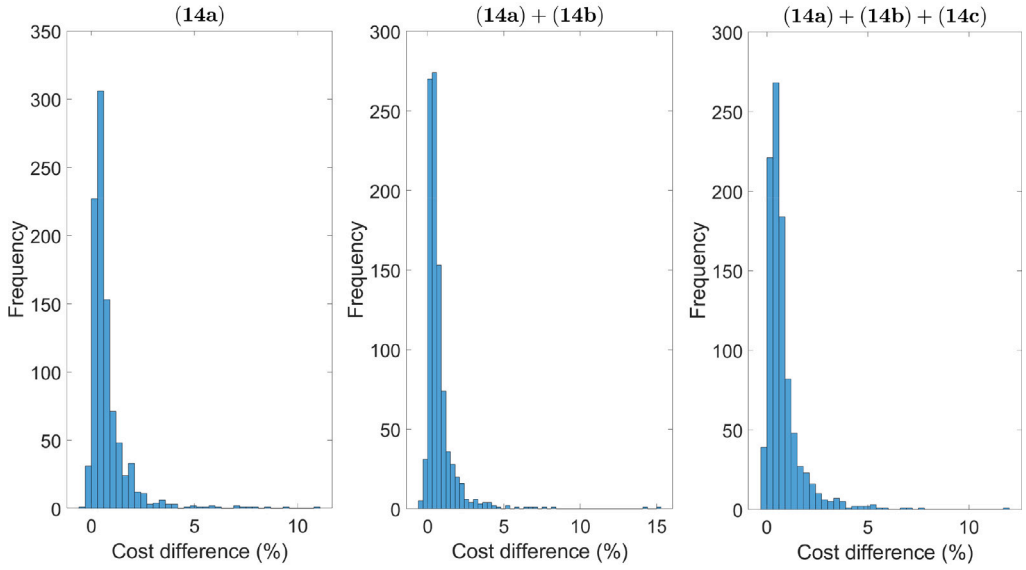


Fig. 5. The histogram of the total cost differences for different cost coefficients and constraints.

parallelization techniques. This enhancement will enable the attainment of high-precision results for complex systems within shorter timeframes. Furthermore, the methodology can be further developed to accommodate high number of agents. Additionally, incorporating multi-objective optimization approaches into this framework offers a promising avenue for further development. We plan to develop the proposed method by addressing these aspects in future research endeavors.

CRediT authorship contribution statement

Burak Dindar: Writing – original draft, Visualization, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Can Berk Saner:** Writing – original draft, Visualization, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Hüseyin K. Çakmak:** Writing – review & editing, Visualization, Validation, Supervision, Project administration, Funding acquisition. **Veit Hagenmeyer:** Writing – review & editing, Validation, Supervision, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was partly conducted within the framework of the Helmholtz Program Energy System Design (ESD) and the DigIPlat project, which received funding in the framework of the joint programming initiative ERA-Net Smart Energy Systems' focus initiative Digital Transformation for the Energy Transition, with support from the European Union's Horizon 2020 research and innovation program under grant agreement No 883973.

Data availability

Data will be made available on request.

References

[1] J. Rahman, C. Feng, J. Zhang, Machine learning-aided security constrained optimal power flow, in: 2020 IEEE Power & Energy Society General Meeting, PESGM, IEEE, 2020, pp. 1–5.
[2] Energy Information Administration (US), Annual Energy Outlook 2012: With Projections to 2035, Government Printing Office, 2012.
[3] Bundesverband der Energie- und Wasserwirtschaft, Redispatch 2.0, 2023, https://www.bdew.de/energie/redispatch-20/.
[4] TransnetBW, Studie zu Redispatch 3.0 vorgestellt, 2022, https://www.transnetbw.de/de/newsroom/pressemitteilungen/studie-zu-redispatch-3-0-vorgestellt.
[5] M. McCulloch, J. Dierenbach, M. Baatar, D. Graeber, K. Tolstrup, S. Fanta, V. Zobernig, How interoperability of flexibility platforms enables market design opportunities, in: 2023 19th International Conference on the European Energy Market, EEM, IEEE, 2023, pp. 1–4.
[6] Bundesverband der Energie- und Wasserwirtschaft, enera, 2020, https://projekt-enera.de/.
[7] T. Mühlpfordt, X. Dai, A. Engelmann, V. Hagenmeyer, Distributed power flow and distributed optimization—Formulation, solution, and open source implementation, Sustain. Energy, Grids Netw. 26 (2021) 100471.
[8] K. Christakou, D.-C. Tomozei, J.-Y. Le Boudec, M. Paolone, AC OPF in radial distribution networks–Part II: An augmented Lagrangian-based OPF algorithm, distributable via primal decomposition, Electr. Power Syst. Res. 150 (2017) 24–35.
[9] X. Dai, Y. Lian, Y. Jiang, C.N. Jones, V. Hagenmeyer, Hypergraph-based fast distributed AC power flow optimization, in: 2023 62nd IEEE Conference on Decision and Control, CDC, IEEE, 2023, pp. 4572–4579.
[10] T.W. Mak, F. Fioretto, P. Van Hentenryck, Privacy-preserving obfuscation for distributed power systems, Electr. Power Syst. Res. 189 (2020) 106718.
[11] M. Ryu, K. Kim, A privacy-preserving distributed control of optimal power flow, IEEE Trans. Power Syst. 37 (3) (2021) 2042–2051.
[12] X. Niu, H.K. Nguyen, J. Sun, Z. Han, Privacy-preserving computation for large-scale security-constrained optimal power flow problem in smart grid, IEEE Access 9 (2021) 148144–148155.
[13] T. Wu, C. Zhao, Y.-J.A. Zhang, Privacy-preserving distributed optimal power flow with partially homomorphic encryption, IEEE Trans. Smart Grid 12 (5) (2021) 4506–4521.
[14] N. Guha, Z. Wang, M. Wytock, A. Majumdar, Machine learning for AC optimal power flow, 2019, arXiv preprint arXiv:1910.08842.
[15] F. Hasan, A. Kargarian, A. Mohammadi, A survey on applications of machine learning for optimal power flow, in: 2020 IEEE Texas Power and Energy Conference, TPEC, IEEE, 2020, pp. 1–6.
[16] M. Chatzos, F. Fioretto, T.W. Mak, P. Van Hentenryck, High-fidelity machine learning approximations of large-scale optimal power flow, 2020, arXiv preprint arXiv:2006.16356.
[17] X. Lei, Z. Yang, J. Yu, J. Zhao, Q. Gao, H. Yu, Data-driven optimal power flow: A physics-informed machine learning approach, IEEE Trans. Power Syst. 36 (1) (2020) 346–354.
[18] X. Pan, DeepOPF: deep neural networks for optimal power flow, in: Proceedings of the 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation, 2021, pp. 250–251.

[19] T.W. Mak, M. Chatzos, M. Tanneau, P. Van Hentenryck, Learning regionally decentralized ac optimal power flows with admm, IEEE Trans. Smart Grid (2023).
[20] I. Mezghani, S. Misra, D. Deka, Stochastic AC optimal power flow: A data-driven approach, Electr. Power Syst. Res. 189 (2020) 106567.
[21] A.-A.B. Bugaje, J.L. Cremer, G. Strbac, Split-based sequential sampling for realtime security assessment, Int. J. Electr. Power Energy Syst. 146 (2023) 108790.
[22] A. Venzke, D.K. Molzahn, S. Chatzivasileiadis, Efficient creation of datasets for data-driven power system applications, Electr. Power Syst. Res. 190 (2021) 106614.
[23] B. Zhang, D. Tse, Geometry of feasible injection region of power networks, in: 2011 49th Annual Allerton Conference on Communication, Control, and Computing, Allerton, IEEE, 2011, pp. 1508–1515.
[24] B.C. Lesieutre, I.A. Hiskens, Convexity of the set of feasible injections and revenue adequacy in FTR markets, IEEE Trans. Power Syst. 20 (4) (2005) 1790–1798.
[25] M. Fischetti, J. Jo, Deep neural networks and mixed integer linear optimization, Constraints 23 (3) (2018) 296–309.
[26] F. Rossi, E.P. Araujo, M.C. Mañe, O.G. Bellmunt, Data generation methodology for machine learning-based power system stability studies, in: 2022 IEEE PES Innovative Smart Grid Technologies Conference Europe, ISGT-Europe, IEEE, 2022, pp. 1–5.
[27] J. Jalving, M. Eydenberg, L. Blakely, A. Castillo, Z. Kilwein, J.K. Skolfield, F. Boukouvala, C. Laird, Physics-informed machine learning with optimization-based guarantees: Applications to AC power flow, Int. J. Electr. Power Energy Syst. 157 (2024) 109741.
[28] D.K. Molzahn, Computing the feasible spaces of optimal power flow problems, IEEE Trans. Power Syst. 32 (6) (2017) 4752–4763.
[29] M. Bichler, J. Knörr, Getting prices right on electricity spot markets: On the economic impact of advanced power flow models, Energy Econ. 126 (2023) 106968.
[30] T. Joswig-Jones, K. Baker, A.S. Zamzam, OPF-learn: An open-source framework for creating representative AC optimal power flow datasets, in: 2022 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference, ISGT, IEEE, 2022, pp. 1–5.
[31] A.-A.B. Bugaje, J.L. Cremer, G. Strbac, Generating quality datasets for real-time security assessment: Balancing historically relevant and rare feasible operating conditions, Int. J. Electr. Power Energy Syst. 154 (2023) 109427.
[32] I.V. Nadal, S. Chevalier, Scalable bilevel optimization for generating maximally representative OPF datasets, 2023, arXiv preprint arXiv:2304.10912.
[33] C. Bingane, M.F. Anjos, S. Le Digabel, Tight-and-cheap conic relaxation for the AC optimal power flow problem, IEEE Trans. Power Syst. 33 (6) (2018) 7181–7188.
[34] S. Babaeinejadsarookolaee, A. Birchfield, R.D. Christie, C. Coffrin, C. DeMarco, R. Diao, M. Ferris, S. Fliscounakis, S. Greene, R. Huang, et al., The power grid library for benchmarking ac optimal power flow algorithms, 2019, arXiv preprint arXiv:1908.02788.
[35] R.D. Zimmerman, C.E. Murillo-Sánchez, R.J. Thomas, MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education, IEEE Trans. Power Syst. 26 (1) (2011) 12–19, http://dx.doi.org/10.1109/TPWRS.2010.2051168.
[36] R.H. Byrd, J. Nocedal, R.A. Waltz, Knitro: An integrated package for nonlinear optimization, Large-Scale Nonlinear Optim. (2006) 35–59.
[37] Gurobi Optimization, LLC, Gurobi Optimizer Reference Manual, 2023, URL https://www.gurobi.com.
[38] J. Löfberg, YALMIP : A toolbox for modeling and optimization in MATLAB, in: Proceedings of the CACSD Conference, Taipei, Taiwan, 2004.
[39] M. Abadi, et al., TensorFlow: Large-scale machine learning on heterogeneous systems, 2015, Software available from tensorflow.org, URL https://www.tensorflow.org/.
[40] F. Chollet, et al., Keras, 2015, https://keras.io.
[41] M. Hossin, M.N. Sulaiman, A review on evaluation metrics for data classification evaluations, Int. J. Data Min. Knowl. Manag. Process. 5 (2) (2015) 1.