Split-based sequential sampling for realtime security assessment

Bugaje, Al Amin B.; Cremer, Jochen L.; Strbac, Goran

DOI
10.1016/j.ijepes.2022.108790

Publication date
2023

Document Version
Final published version

Published in
International Journal of Electrical Power and Energy Systems

Electrical Power and Energy Systems 146 (2023) 108790

Contents lists available at ScienceDirect

International Journal of Electrical Power and Energy Systems

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

Split-based sequential sampling for realtime security assessment

Al-Amin B. Bugaje [a], Jochen L. Cremer [b,*], Goran Strbac [a]

[a] Department of Electrical & Electronic Engineering, Imperial College London, London, SW7 2AZ, UK
[b] Department of Electrical Sustainable Energy, TU Delft, Mekelweg 5, 2628 CD Delft, Netherlands

ARTICLE INFO

Keywords:
Sampling
Database generation
Security assessment
Machine learning
Power system operation

ABSTRACT

Machine learning (ML) for real-time security assessment requires a diverse training database to be accurate for scenarios beyond historical records. Generating diverse operating conditions is highly relevant for the uncertain future of emerging power systems that are completely different to historical power systems. In response, for the first time, this work proposes a novel split-based sequential sampling approach based on optimisation that generates more diverse operation scenarios for training ML models than state-of-the-art approaches. This work also proposes a volume-based coverage metric, the convex hull volume ($\mathcal{CHV}$), to quantify the quality of samplers based on the coverage of generated data. This metric accounts for the distribution of samples across multidimensional space to measure coverage within the physical network limits. Studies on IEEE test cases with 6, 68 and 118 buses demonstrate the efficiency of the approach. Samples generated using the proposed split-based sampling cover 37.5% more volume than random sampling in the IEEE 68-bus system. The proposed $\mathcal{CHV}$ metric can assess the quality of generated samples (standard deviation of 0.74) better than a distance-based coverage metric which outputs the same value (standard deviation of < 0.001) for very different data distributions in the IEEE 68-bus system. As we demonstrate, the proposed split-based sampling is relevant as a pre-step for training ML models for critical tasks such as security assessment.

1. Introduction

The integration of more renewable energy introduces a high level of uncertainty in power systems operations. This uncertainty challenges future reliability management [1]. Conventional reliability management involves considering large safety margins for system operations. Subsequently, the security of the energy supply is reliable in most cases. However, allowing such large safety margins on top of the increasing uncertainty of system operations implies inefficiently utilising the infrastructure at most times. Maintaining this inefficient status-quo requires expensive infrastructure investments in the future. A more efficient approach would be to use the existing infrastructure more exhaustively by lowering safety margins. Nevertheless, this approach needs to improve the situational awareness of the system operation.

The availability of real-time operation data, for instance, from Phasor measurement units (PMUs) [2] allows for carrying out state estimation [3] and the subsequent dynamic security analysis (DSA) of the system. However, with current operating tools and uncertain operation data, it is computationally expensive to assess system security in real-time for numerous possible operating scenarios and disturbances. For example, a single assessment can take up-to 56 s in large systems [4], and several thousand assessments would be needed. In such scenarios, ML is particularly promising as it allows for predictions in real-time with little computational time [5]. A functioning ML-based DSA tool has the potential to increase situational awareness, support reliability management, improve uncertainty handling, and efficiently integrate more renewable energy.

1.1. ML approach to security assessment

The idea of ML for DSA is the following: Offline, a database of many possible operating conditions (OCs) with the corresponding information on whether the OC is secure (or not) for one (or multiple) disturbances is generated. Subsequently, models are trained using ML approaches. Then, in real-time operations, the learned models predict the level of security as an output where the input is the current OC. Although the models may sometimes be inaccurate, the main benefit is that OCs that were not part of the database can be input, and most importantly, no computational time (e.g., for simulations) is needed in real-time operations. This approach is promising to increase the situational awareness of the system operator (SO) by allowing the SO to consider a large number of possible OCs and disturbances in real-time. In light of this development, machine-learned models including decision trees (DTs) [6,7], support vector machines (SVMs) [8], and

* Corresponding author.
E-mail addresses: abb18@imperial.ac.uk (A.-A.B. Bugaje), j.l.cremer@tudelft.nl (J.L. Cremer), g.strbac@imperial.ac.uk (G. Strbac).

https://doi.org/10.1016/j.ijepes.2022.108790
Received 23 January 2022; Received in revised form 10 August 2022; Accepted 10 November 2022
Available online 23 November 2022

Nomenclature

Indices

ρ	user-defined threshold on the size of Ω^G
d	index of sample where the primary variable is located
k	index of gaps
N	number of selected secondary variables
n	number of samples that are vertices of the convex hull
p	index of all variables
q	index of selected secondary variables
S	user-defined number of samples to generate
u	index of samples
v	index of all secondary variables

Sets

$\hat{\Omega}$	subset of variables to compute volume
Ω^P	set of all variables
Ω^S	set of generated feasible samples
Ω^G	set of all gaps
Ω^{Q*}	set of selected secondary variables
Ω^Q	set of all secondary variables
$\Omega^{S''}$	set of all generated samples
$\Omega^{S'}$	set of generated infeasible samples
Ω_p^{V-S}	set of ordered generated feasible samples in variable p

Parameters

α	tolerance parameter on variable p in optimisation
β	share of infeasible samples
$\Delta x_p^{(max)}$	maximum gap in variable p
$\Delta x_p^{(u)}$	gap between $(u+1)$th and (u)th largest samples in variable p
ε	weighted Euclidean distance between optimised sample x^* and corresponding target T_p
N_+	number of insecure operating conditions
N_-	number of secure operating conditions
r_p	range of variable p
T_p	target of variable p
w_p	weight of variable p in optimisation
x_p^{LB}	lower bound of variable p
x_p^{UB}	upper bound of variable p

Variables

δ	slack variable in optimisation
x	variable defining an operating condition

Others

γ_i	minimum distance between sample x_i and other samples in Ω^S
$\hat{CHV}$	normalised convex hull volume
λ	coverage metric based on distance
λ	real number
$\mathcal{CHV}$	convex hull volume
$\mathcal{COV}$	distance based coverage metric
$\mathcal{I}(p)$	position of x^* in Ω_p^{V-S}
$\mathcal{MA}$	moving average
$\mathcal{P}(k)$	mapping to retrieve variable index p
$\mathcal{U}(k)$	mapping to retrieve sample index u
C	convex hull of a set of points
f	function describing random selection of Ω^{Q*}
$g(x^*) \leq 0$	constraints on optimised operating condition x^*
h	coverage metric based on point norm distribution
h_i	maximum distance between sample x_i and vertices of its associated cell V_i
V_i	Voronoi tessellation formed by sample x_i
x^*	an optimised operating condition
y	vertex of Voronoi cell

more recently deep learning models [9] have shown promise to assess dynamic stability problems ranging from voltage stability [7,10], transient stability [6,9] and frequency stability [11,12]. Recent works [13–16] show real promise for real-time probabilistic DSA including considering topological changes [17,18]. There, using ML, estimating the dynamic security boundary [19] particularly works well for future low inertia grids [20] and may become possible through the increasing availability of larger amount of (PMU-) data and monitoring tools in control centers [21]. However, as these ML-based approaches to DSA are data-driven these approaches can only be as good as their training database. Using only historical observations as training data is insufficient [22,23]. Therefore, the generation (sampling) of data is highly relevant for the success of all aforementioned approaches [24]. Many papers that focus on developing ML-based models consider the generation of the data, a few have focused only on the generation of database. This work focuses only on the generation of training data.

1.2. Sampling approaches

The prediction performance of ML models is generally a reflection of the quality (coverage, variability, and balance) of the data used in training [25]. The choice of training databases in DSA application differs from other ML applications that use (recorded) observations. Using recorded, historical data for DSA has limitations. Often, the majority of historical observations are secure. However, a good training database needs to consider both secure and insecure conditions [5]. Also, historical OCs rarely involve extreme operating scenarios. Hence, sampling approaches are used to generate synthetic OCs. When generating synthetic samples, firstly, an OC is sampled, then it is assessed with a time-domain simulation for the considered contingency.

The generation of data for ML-based DSA is highly relevant which is why many contributions were made along three types of approaches: the first type of approach, *historic sampling* uses historical records [23], fits a probability distribution to it (e.g., vine-copulas in [26,27] to capture the dependencies between loads and wind power outputs), then generates OCs using Monte-Carlo (MC) type samplings [28,29]. This type of approach is suitable to sample OCs following the same distribution as historical observations. Another variant of *historic sampling* determines the 'relevant' buses to obtain sparse PMU measurements. By selecting subsets of these 'relevant' buses for sampling, the issue of high dimensionality can be mitigated as only a smaller ('relevant') dimension of variables need to be sampled as shown in [30,31]. However, future OCs may be different than historical OCs, and sampling from distributions is unsuitable for creating extreme OCs typically found at the tails of distributions. The second type of approach, *importance sampling* is where the sequence of sampling and classifier training iteratively repeats to maximise high information content [32–36]. In each iteration, the sampling (e.g., with MC-sampler) generates possible OCs. Then, the classifier quantifies the importance of these OCs based on the predicting confidence. Subsequently, the security assessment is

used only on samples with low confidence. For instance, Yan [22] uses entropy as a metric to generate 'relevant' samples closer to the decision boundary. [24] uses 'directed walk' methods to samples around the decision boundary. The third type of approach, *generic sampling*, generates points uniformly distributed in the feasible space to explore all possible OCs. However, large systems require large amounts of generated data, and most data adds little knowledge to the database. For instance, Jafar [37] uses the Latin hypercube sampling (LHS) approach to uniformly sample the entire search space, and researchers in [38] sample within the feasible neighbourhood of OCs, while researchers in [39] proposed an outer approximation to convexify the original non-convex feasible space, then sample from the convex region to generate samples close to the security boundary. Venzke [40] uses infeasibility certificates based on separating hyperplanes to discard large portions of the input space as infeasible. More recently, the authors in [41] developed a framework to generate representative samples that span the AC OPF feasible space by uniformly sampling loads from a convex input space and using infeasibility certificates to reduce the search space. The drawback of the first and second type of sampling approaches is that they neglect some feasible OCs. The first approach is biased towards historical observations, and the second towards the importance of learning the security boundary. Hence, sampling extreme OCs with those approaches is rare. However, studying extreme OCs beyond historical records is crucial as these can be dangerous for system operations. The challenge of the third type of approach is that sampling in high-dimensions is not trivial. Therefore, a current research gap and need is an efficient *generic sampling* approach that scales to larger systems and can generate extreme synthetic OCs as the introduction of intermittent renewables into the energy mix means that the power system will experience new OCs that were historically not covered.

Other fields faced with similar sampling challenges from large solution spaces have proposed novel approaches. In particular, bio-engineering employs random sampling $\mathcal{RS}$ techniques to investigate constraint-based metabolic reactions that have a large solution space. A popular sampling technique often employed is the family of "hit-and-run" (HR) samplers (ACHR [42], CHRR [43]) that randomly choose directions to traverse a model's solution space based on warm start positions. This approach relies on the convexity of the solution space and requires relaxation of non-convex models as found in the II-ACHR sampler [44]. A new approach called `GAPSPLIT` was introduced to sample models directly [45]. The sampler generates points by jumping to unexplored regions of the space in contrast with the random walk approach employed by HR samplers. `GAPSPLIT` is a competitive alternative to HR samplers that scales relative to the size of the model and can sample directly from non-convex models. Samples generated with `GAPSPLIT` also have better coverage than ACHR and CHRR on unbounded model variables. The approach was used in [46] to sample a highly constrained solution space.

The state-of-the-art methods in the literature focusing on generating data for a training database of ML-based DSA is presented in Table 1. Our proposed approach is fundamentally novel to other peer-reviewed works we have investigated. Specifically, our approach is novel in the way the initial OCs are being sampled. Our approach conceptually outperforms other state-of-the-methods in it's practicality, and ability to generate all (feasible) possible operating conditions. Typically, most state-of-the-art works consider generator outputs to be scheduled to represent conventional systems operations, often as a result of solving the optimal power flow problem that minimises generation cost [26,32,41,33,34,31,48,36]. However, as it is likely that the initial OC where a fault occurs is different from the optimal set-points, it is necessary to develop methods that explore these likely OCs [22]. Thus, a first point of comparison to generate pre-fault OCs is with methods that consider the OPF to generate initial OCs. The other approaches in the literature explore the entire feasible space in a generic way via random sampling, often using the Latin Hypercube sampling to generate initial OCs [37,39,40,24,22]. As a consequence, a second and more pivotal comparison is with those methods that aim to uniformly cover the search space using techniques like the Latin Hypercube sampling. As highlighted in Table 1, additionally, a major shortcoming of *historical* and some *importance sampling* approaches is that the resulting database of OCs represents only a small portion of the feasible space. Consequently *generic sampling* allows the exploration of the full physical feasible space. While existing generic sampling approaches currently in the literature attempt to discard sections of the search space via rapid rejection sampling [41,39,40,24], our proposed method differs fundamentally from state-of-the-art approaches by optimally exploring the feasible space in an iterative fashion. This exploration is done by varying the objective function and active constraints while respecting all the physical feasible constraints. The novelty of this work stems from presenting for the first time a *generic sampling* method that categorically explores the feasible space in an optimal manner. Finally, our proposed approach is versatile and could be further developed towards the combination with other database generation approaches like importance sampling and together with historical records. In parallel to this work, the *generic sampling* approach [47] investigates multiple objective functions to explore the feasible space.

1.3. Measuring quality of sampling

The quality of a training database is a measure of coverage of the feasible space and data usefulness, representing the pre- and post- fault data, respectively, in ML DSA application. For the coverage, typically, a set of points is said to uniformly cover a region when the points satisfy the following characteristics: (1) placed equidistant relative to one another (2) cover the entire region/volume of interest (3) distributed equally along all directions [49]. Point-to-point coverage measures focus on the first characteristic and aim to quantify how well the points are placed relative to one another. Examples of such metrics include the coverage metric (COV) used in [45], the coefficient of variation (λ) and mesh ratio (γ). Volumetric coverage measures, however, combine the first characteristic with one or both of the other two. Examples of volumetric measures based on Voronoi tessellation include point norm distribution (h), point distribution ratio (μ), regularity metric (χ), etc. [49]. In high dimensional space, proximity measures (point-to-point coverage measures) used in two or three-dimensional space do not carry the same intuitive descriptive information quality [50]. The intuition of Euclidean distance falls apart, and a skin-effect-like tendency is observed such that the volume is concentrated around the skin of a high dimensional hyper-sphere instead of the centre [51]. Due to this concentration effect, the relative contrast between far and near points diminishes as the dimensionality increases, making it difficult to discriminate between far and near points [52]. Therefore, a current research gap and need is a metric that can quantify the quality of a training database for *generic sampling* approaches that have the objective to generate diverse OCs in the physical feasible space for power system DSA application.

For data usefulness, typically the issue of imbalanced datasets in the post-fault label is in focus. In DSA application, the distribution of secure/insecure OCs represents an important consideration for training ML models. There, the state-of-the-art methods in the literature use a preprocessing step such as synthetic minority oversampling (SMOTE) [53] and adaptive synthetic sampling (ADASYN) [54] to achieve this balance, usually to supplement with insecure OCs. The second way to address this imbalance is by combining historical records with OCs generated using a *generic sampling* approach. In power system security assessment, accurately predicting insecure OCs is more important than predicting secure ones [14], and as historical OCs are disproportionately biased with more secure OCs, it motivates the creation of synthetic datasets.

This work focuses on the first quality measure of coverage and aims to generate pre-fault data so that the datasets can have diverse OCs. The motivation of this work is to fully concentrate on the issue of variability in the pre-fault database, where this does not include variability of the post-fault label.

Table 1
Summary of relevant state-of-the-art works on database generation for ML-based DSA.

Reference	Type	Sampling of initial OCs	Advantages	Shortcomings
[41]	Generic sampling	Solving OPF to minimise generation cost	Uses convex relaxations and hyperplanes to discard large sections of the input space. Explores load space via Monte-Carlo sampling	Only considers generator outputs obtained from solving OPF to represent conventional operation, which is a small subset of the feasible space.
[40]	Generic sampling	LHC sampling or uniform sampling.	Systematically covers the search space with uniform sampling while discarding large hyperplanes of infeasible regions	Fitting a multivariate distribution around secure OCs only generates OCs of similar distribution without exploring other possibilities.
[39]	Generic sampling	LHC sampling	Systematically covers the search space with uniform sampling while discarding hyperspheres of infeasible regions	Discarding hyperspheres of many initialisation points in high-dimensions in not computationally trivial
[33]	Importance sampling	Solving OPF to minimise generation cost	Focuses on sampling close to the decision boundary thereby reducing computational budget	Biassing the sampling towards the security boundary ignores rare extreme OCs. Only considers OPF solutions
[26]	Historical sampling	Solving OPF to minimise generation cost	High density sampling of OCs from historical records.	Neglects unseen or rare OCs that are critical to be analysed.
[24]	Generic + Importance sampling	Grid search, uniform sampling in each dimension or LHC sampling	Focuses on sampling close to the decision boundary using enhancement methods such as directed walks, the prediction model as a pre-selection tool for relevant samples and performance guarantee of entire regions.	Relies on resampling techniques to bias sampling to narrow regions of the space. Using performance guarantees significantly reduces the search space and can affect model performance.
[47]	Generic sampling	Sequentially generated to explore the physical feasible space	Sequentially explores the physical feasible space to maximise distance from previously generated samples.	Performance on larger systems (bus $\geq$ 68) is not tested.
[34]	Importance sampling	Solving OPF to minimise generation cost	Computes quadratic approximation of the security boundary and use importance sampling to generate OCs	Dataset represents only a small portion of the feasible space.
[32]	Importance sampling	Solving OPF to minimise generation cost	Identifies the decision boundary and fits a polynomial function so as to sample OCs within the proximity of the boundary	Dataset represents only a small portion of the feasible space
[22]	Importance sampling	LHC sampling	Uses a transient stability index to direct sampling in a high-information content region formulated as an optimisation problem.	Focuses only on generating datasets for identifying the transient stability boundary.
[30]	Historical sampling	Solving OPF to minimise generation cost	Dimensionality reduction using neural networks reduces the search space considerably thereby improving computational time.	A large part of the search space is ignored. Rare OCs are not considered.
[37]	Importance sampling	LHC sampling	Considers rare cases by fitting a generalised pareto distribution to the tail-region.	Dataset represents only a small portion of the feasible space as only OPF solutions are considered.
[31]	Historical sampling	Solving OPF to minimise generation cost	Uses advancements in GAN to address the issue of missing PMU data when implementing ML-based DSA.	Dataset represents only a small portion of the feasible space as only OPF solutions are considered. Method cannot generate arbitrarily new OCs.
[35]	Importance sampling	Solving OPF to minimise generation cost	Interpolating between secure and insecure cases to sample new points ensures the creation of relevant samples	Dataset represents only a small portion of the feasible space.
[48]	Historical sampling	Solving OPF to minimise generation cost	Adopts a feature selection strategy to optimise PMU data collection for fast and robust prediction.	Dataset represents only a small portion of the feasible space as only OPF solutions are considered. Method cannot generate arbitrarily new OCs.
[23]	Historical sampling	Historical records.	Use of a cycleGAN model to refine simulated data such that it mimics actual transients from historical data improves the quality of synthetic data.	Dataset represents only a small portion of the feasible space as only historical records are considered.
[36]	Importance sampling	Solving OPF to minimise generation cost	Improves computation time needed to build a transient stability assessment database using a semi-supervised ensemble learning approach.	Dataset represents only a small portion of the feasible space as only OPF solutions are considered.
Proposed approach	Generic sampling	Sequentially generated to explore the physical feasible space	Sequentially explores the physical feasible space to maximise distance from previously generated samples.	Method does not currently consider class imbalance in the formulation

1.4. Contributions

This work proposes a novel split-based *generic sampling* approach, GAPSPLIT*. This novel split-based approach is a modification of the GAPSPLIT approach. The novel split-based sampling approach aims to systematically generate diverse pre-fault operating conditions. The proposed approach covers previously unexplored OCs that are physical feasible but have not occurred in the past. With the proposed approach, high-quality databases (of pre-fault OCs) can be generated for training ML models used in real-time DSA. The proposed algorithm's crucial advantage over other statistical, distribution-based approaches that require fitting to a pre-existing database is the ability to consider the full physical search space defined by the ACOPF and requires no historical data to work. In this paper, the contribution is threefold: first,

for the first time, this work investigates the GAPSPLIT approach for power system application. Second, this work modifies the GAPSPLIT approach to make it suitable for power system application. Third, this work investigates metrics to assess the quality of a *generic sampling* approach.

In the first contribution, the GAPSPLIT approach uses mathematical optimisation for sampling [45]. In this proposed work, sampling feasible OCs considers all power system constraints, such as power flow equations, line flow constraints, and node balances from the Alternating Current (AC) model. Then, an optimisation is solved sequentially for each new sample. Each sequence considers previously generated samples and determines the maximal gap in the entire feasible region, then uses optimisation to add physical constraints at the maximal gap, which is the sampling target. The approach considers primary and secondary targets, where the primary target is a hard-constraint on the maximal gap in the optimisation, and the secondary targets are in the objective function to minimise the Euclidean distance to the target.

In the second contribution, the proposed modification from GAPSPLIT to the proposed GAPSPLIT* approach has two pivotal advancements: to avoid converging to infeasible samples that do not satisfy the power flow equations and to efficiently analyse previously generated data to boost scalability to larger systems. The first advancement to avoid infeasibility is achieved by one of two proposed approaches: (i) relaxing the hard constraint on the primary target and activating only the constraints on secondary targets and (ii) storing infeasible samples and considering them as closed gaps to prevent the sampling from diverging. The second advancement to support scalability to larger systems is achieved by introducing the efficient sorting of sets.

In the third contribution, this work proposes a new volumetric coverage assessment metric, the convex hull volume ($\mathcal{CHV}$) to assess the quality of a *generic sampling* approach. The convex hull is the union of all simplices with vertices in a set, i.e., the smallest convex polygon that surrounds a set of points. The ($\mathcal{CHV}$) of this envelope serves as a metric to represent the coverage of points. In our studies, we show the benefits of $\mathcal{CHV}$ as a better coverage metric to distance-based coverage metrics.

The rest of the paper is structured as follows: Section 2 discusses the regular split-based sampling GAPSPLIT and the proposed modified split-based approach GAPSPLIT*. Section 3 introduces performance measuring metrics, including coverage using the proposed $\mathcal{CHV}$ metric. Section 4 outlines case studies to compare the performance of our proposed modified sampling approach, the proposed performance metric and the computational performance. Tests are carried out on the IEEE 6-bus, the IEEE 68-bus, and the IEEE 118-bus systems. Section 5 concludes the paper.

2. Split-based sampling

The proposed split-based approach follows the idea of *generic sampling* that aims to uniformly cover the full physical feasible space with all possible OCs.

2.1. Regular split-based approach

This section describes the GAPSPLIT sampling algorithm [45] that allows to include physical model-based constraints by formulating the sampling as an optimisation problem. *Algorithm 1* illustrates this sampling strategy that comprises an initialisation step, an iteration step that performs analysis and optimisation, and criteria to stop the iterating algorithm.

The algorithm initialises with an empty set of samples $|\Omega^S| = 0$. The symbol $|\cdot|$ denotes the cardinality of a set. The subsequently generated samples Ω^S have a sample vector $x \in \mathcal{R}^{|\Omega^P|}$ that describes the OC of the power system in $|\Omega^P|$ dimensions and satisfies the constraints of the physical model. The lower and upper bounds of the pth variable

Algorithm 1: GAPSPLIT algorithm

Define samples-set Ω^S with each sample $x \in \mathcal{R}^{|\Omega^P|}$;
Define range $r_p = x_p^{UB} - x_p^{LB} \quad \forall\, p \in \Omega^P$;
Define normalisation parameter $w_p = \frac{1}{r_p^2}\ \forall\, p \in \Omega^P$;
while *true* **do**
  Sort $\Omega^S \quad \forall\, p \in \Omega^P$;
  Compute $\Delta x_p^{(u)} = x_p^{(u+1)} - x_p^{(u)}\ \forall\, p \in \Omega^P$;
  Search $\forall\, p \in \Omega^P$
  $\Delta x_p^{(max)} = \max\{\Delta x_p^{(k)} \mid \forall\, k = 1, 2, \ldots, (|\Omega^S| + 1)\}$;
  Select $\tilde{p}$ s.t $x_{\tilde{p}}^{(max)} = \max\{\Delta x_p^{(max)} \mid \forall\, p \in \Omega^P\}$;
  Compute $T_p = \frac{x_p^{(d+1)} - x_p^{(d)}}{2} + x_p^{(d)}\ \forall\, p \in \Omega^P$;
  Select $\Omega^{Q^*} = \{q \mid q = f(v), f : [N] \mapsto \Omega^Q, N \leq |\Omega^Q|\}$;
  Solve ;

$$\min_{x^*} \sum_{p \in \Omega^{Q^*}} w_p (x_p^* - T_p)^2$$

$$g(x^*) \leq 0$$

$$(1-\alpha) T_{\tilde{p}} \leq x_{\tilde{p}}^* \leq (1+\alpha) T_{\tilde{p}}$$

  Update $\Omega^S \longleftarrow x^*$;
  Recalculate $\Delta x_p^{(max)}\ \forall\, p \in \Omega^P$;
  if $|\Omega^S| \leq S$ **then**
    return Ω^S;
  end
end

are denoted as x_p^{LB} and x_p^{UB}, respectively, and the ranges are $r_p = x_p^{UB} - x_p^{LB}$.

In each iteration of GAPSPLIT, the algorithm generates a single optimised sample x^*, starting with an analysis of previous samples Ω^S. The analysis begins with sorting the samples Ω^S for each variable $p \in \Omega^P$ resulting in $|\Omega^P|$ ordered sets of the same samples Ω^S. GAPSPLIT sorts these sets according to the values in variable p

$$\Omega_p^{V-S} = \{x_p^{(u)} \mid \forall\, u = 1, 2, \ldots, |\Omega^S|,\ x_p^{(u+1)} \geq x_p^{(u)}\} \tag{1}$$

where $x_p^{(u)}$ corresponds to the uth largest sample in the pth variable. Subsequently, the algorithm computes the gaps of the samples next to each other

$$\Delta x_p^{(u)} = x_p^{(u+1)} - x_p^{(u)} \tag{2}$$

and then identifies the maximal gap in each pth variable

$$\Delta x_p^{(max)} = \max\{\Delta x_p^{(k)} \ \mid\ \forall\, k = 1, 2, \ldots, (|\Omega^S|)\}, \tag{3}$$

where the algorithm denotes the two samples next to the maximal gap $\Delta x_p^{(max)} = x_p^{(d+1)} - x_p^{(d)}$ with $(d+1)$ and (d). The maximal gap among all variables is

$$\Delta x_{\tilde{p}}^{(max)} = \max\{\Delta x_p^{(max)} \ \mid\ \forall\, p = 1, 2, \ldots, |\Omega^P|\}, \tag{4}$$

where $\tilde{p}$ denotes the variable with the maximal gap called the primary variable. All other variables are called secondary variables $\Omega^Q = \Omega^P \setminus \tilde{p}$. Subsequently, the algorithm computes targets for all primary and secondary variables at the centre of their respective maximal gaps

$$T_p = \frac{x_p^{(d+1)} - x_p^{(d)}}{2} + x_p^{(d)}; \tag{5}$$

these are accordingly called primary and secondary targets, e.g., the primary target is $T_{\tilde{p}}$. Subsequently, the algorithm considers a subset of secondary variables $\Omega^{Q^*} \subset \Omega^Q$ as not all secondary variables are further

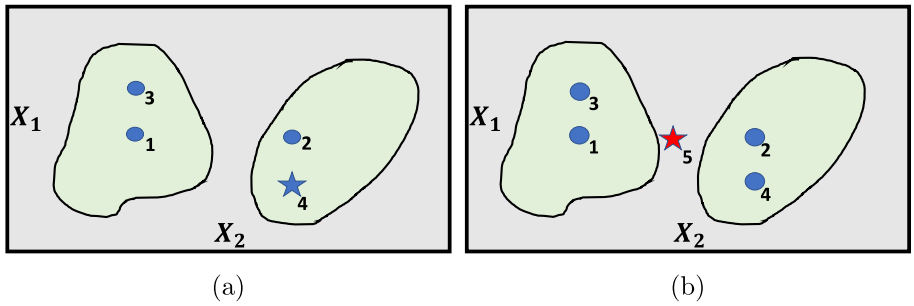


Fig. 1. (a) GAPSPLIT generates new samples by attempting to split maximal gaps. (b) GAPSPLIT converges to an infeasible sample when the primary target (★) is located in the infeasible region. Max gap is found in X_2.

Input space Feasible space Infeasible region.

needed. There are multiple user-specific ways to select the subset of secondary variables Ω^{Q^*}. One way is to consider a random selection

$$\Omega^{Q^*} = \{q \mid q = f(v), f : [N] \mapsto \Omega^Q, |f| = N, N \leq |\Omega^Q|\} \tag{6}$$

of a subset of N elements from Ω^Q, where N is fixed and defined by the user (e.g., $N = 0.05\,|\Omega^Q|$). Then, the algorithm uses random selection $f : [N] \mapsto \Omega^Q$ in each iteration. Other ways to select the secondary variables are in [45], including to select $|\Omega^{Q^*}| = 0$ as empty. In the remainder of the text, referring to secondary variables and targets corresponds to the subset of secondary variables Ω^{Q^*}.

After the above analysis, the GAPSPLIT algorithm generates a single, new sample with the mathematical optimisation

$$\begin{aligned} \underset{x^*}{\text{minimise}} \quad & \sum_{p \in \Omega^{Q^*}} w_p (x_p^* - T_p)^2 \\ \text{subject to} \quad & g(x^*) \leq 0 \\ & (1-\alpha) T_{\tilde{p}} \leq x_{\tilde{p}}^* \leq (1+\alpha) T_{\tilde{p}}, \end{aligned} \tag{7}$$

where the optimisation considers a constraint on the value $x_{\tilde{p}}^*$ of the primary variable $\tilde{p}$ at the primary target $T_{\tilde{p}}$ with a relaxation to avoid numerical issues. The relaxation is considered with a tolerance parameter α on the primary target $T_{\tilde{p}}$ (e.g., of $\alpha = 0.001$). This optimisation minimises the mean squared error from the generated sample x^* to the targets of the selected secondary variables $\Omega^{Q^*} \subset \Omega^Q$. $w_p = \frac{1}{r_p^2}$ is a normalisation parameter that re-weighs all variables equally. $g(x) \leq 0$ are the constraints that define the feasible space including power system constraints such as power flow equations. This optimisation aims to consider the physical constraints of the power system and to split the gaps between the previously generated samples (that is why the algorithm is called GAPSPLIT). The optimisation in (7) returns a new sample, the optimised OC x^*. This sample x^* is added to the set of samples $\Omega^S \leftarrow x^*$, and the next iteration is started.

The algorithm terminates when a user-specified criterion is met, for example, when a specified number of samples S have been generated. Then, a new sample is only generated if $|\Omega^S| \leq S$, otherwise the sampling algorithm stops.

2.2. Issues with GAPSPLIT

Two issues arise when using the above split-based approach for sampling power system OCs. The first issue is the low coverage of the physical feasible space and the second issue is the computational inefficiency when generating a large number of samples.

The first issue of low coverage is the result of GAPSPLIT converging to infeasible regions. GAPSPLIT may converge to such infeasible regions when the search space (feasible space) is non-convex and disconnected as in power systems. The feasible space in power systems is the set of OCs that satisfy all operational equality and inequality constraints in $g(x) \leq 0$ [55]. When the GAPSPLIT algorithm locates the primary target $T_{\tilde{p}}$ in an infeasible region, the algorithm results in an infeasible optimisation (7) and returns an infeasible sample x^* as the constraint $g(x) \leq 0$ when $\lim_{x_{\tilde{p}} \to T_{\tilde{p}}} g(x) > 0$ is not met. This issue is illustrated in Fig. 1. In Fig. 1(a), the primary target $T_{\tilde{p}}$ of the 4th candidate sample is in the feasible space, and GAPSPLIT successfully generates a corresponding sample. However, when trying to generate the following 5th candidate sample in Fig. 1(b), the primary target $T_{\tilde{p}}$ is in the infeasible region where $\lim_{x_{\tilde{p}} \to T_{\tilde{p}}} g(x) > 0$. Hence, GAPSPLIT is unable to generate the 5th candidate sample as $\lim_{x_{\tilde{p}} \to T_{\tilde{p}}} g(x) > 0$. As a result, the maximal gap $\Delta x_{\tilde{p}}^{(max)}$ from Eq. (4) does not change in subsequent iterations as all the gaps $\Delta x_p^{(u)}$ between previously generated samples remain unchanged, and consequently, GAPSPLIT converges to that infeasible sample (5th candidate sample in Fig. 1(b)). The second issue is the computational bottleneck of sequential sampling approaches in high-dimensional settings. This bottleneck is particularly critical in power systems that have a large number of variables $|\Omega^P|$ and require a large number of samples $|\Omega^S|$. The computational bottleneck of some sequential sampling approaches, such as GAPSPLIT, is that they often need to analyse a large number of previously generated samples Ω^S in each iteration. Table 2 analyses the computational requirements in each iteration for the GAPSPLIT algorithm in Big-$\mathcal{O}()$ notation to demonstrate this issue. In each iteration, the sorting of samples Ω^S, computing of gaps $\Delta x_p^{(u)}$ and maximal gaps $\Delta x_p^{(max)}$ steps have complexities of $\mathcal{O}(|\Omega^P||\Omega^S|\log|\Omega^S|)$, $\mathcal{O}(|\Omega^P||\Omega^S|)$ and $\mathcal{O}(|\Omega^P||\Omega^S|)$ respectively. The key bottleneck is the sorting step which grows $\mathcal{O}(|\Omega^P||\Omega^S|\log|\Omega^S|)$ as the size of $|\Omega^S| \to a$, where $a \gg 1$ is a large number.

Table 2
Computational analysis of the GAPSPLIT algorithm where $|\Omega^P|$ is a constant and $|\Omega^S|$ corresponds to the number of iterations. The sorting step is the computational bottleneck of the algorithm.

Steps		Computation time
Sort $\Omega_p^{V-S}\ \forall\, p$	in Eq. (1)	$\mathcal{O}(\lvert\Omega^P\rvert\lvert\Omega^S\rvert\log\lvert\Omega^S\rvert)$
Compute $\Delta x_p^{(u)}$	in Eq. (2)	$\mathcal{O}(\lvert\Omega^P\rvert\lvert\Omega^S\rvert)$
Compute $\Delta x_p^{(max)}$	in Eq. (3)	$\mathcal{O}(\lvert\Omega^P\rvert\lvert\Omega^S\rvert)$
Compute $x_{\tilde{p}}^{(max)}$	in Eq. (4)	$\mathcal{O}(\lvert\Omega^P\rvert)$

2.3. Proposed split-based approach: GAPSPLIT*

Our proposed GAPSPLIT* approach improves the GAPSPLIT approach with two modifications to address each of the above issues as follows.

2.3.1. Exclusive sampling of secondary variables and introducing the set of infeasible samples

The first proposed modification of GAPSPLIT* approach is twofold (i) sampling exclusively with secondary variables Ω^{Q^*} and (ii) considering infeasible samples in the subsequent progressions of GAPSPLIT* algorithm. This modification addresses the first issue of low coverage

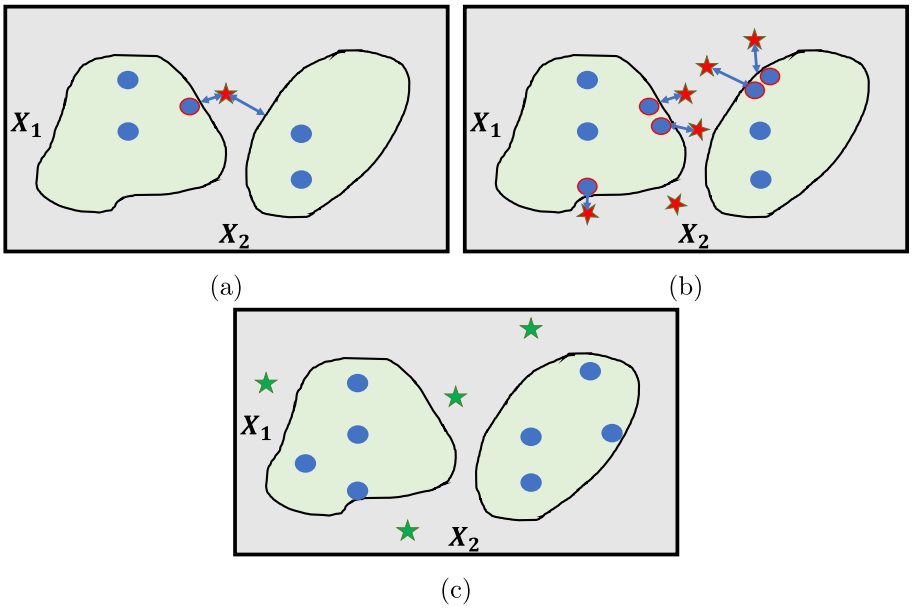


Fig. 2. (a) GAPSPLIT* uses only secondary targets (★) to minimise the shortest distance to the feasible region. (b) GAPSPLIT* with only secondary targets (★) generates samples around the boundary of the feasible space (●) when T_p is located in infeasible regions. (c) GAPSPLIT* re-directs sampling to other regions of the physical space by memorising infeasible targets (★)

when the feasible space is non-convex and disconnected as in power systems.

The modification (i) of sampling exclusively with secondary variables Ω^{Q^*} is to discard the hard constraint on the primary variable $\tilde{p}$ in optimisation (7). Therefore, the optimisation simplifies to

$$\begin{aligned} &\underset{x^*}{\text{minimise}} \quad \sum_{p\in\Omega^{Q^*}} w_p(x_p^* - T_p)^2 \\ &\text{subject to} \quad g(x^*) \le 0, \end{aligned} \tag{8}$$

where the objective is to minimise the mean squared error of x^* to secondary targets T_p, $\forall\ p \in \Omega^{Q^*}$ of the secondary variables. The effect is illustrated in Fig. 2(a) where a target is located in the infeasible region where $\lim_{x_p \to T_p} g(x) > 0$. When comparing Fig. 1(b) (GAPSPLIT) with Fig. 2(a) (GAPSPLIT*), GAPSPLIT would converge to an infeasible sample when the primary target $T_{\tilde{p}}$ is located in the infeasible region. However, GAPSPLIT* addresses this issue by removing the hard constraint on $\tilde{p}$ and minimising the distance to the optimised feasible OC, marked with a red circle in Fig. 2(a). This minimisation of distances generates samples around the boundary of the feasible space and close to each other as illustrated in Fig. 2(b). This accumulation does not support effectively covering the full feasible space.

The modification (ii) addresses the issue of accumulating infeasible samples presented in Fig. 2(b). This modification (ii) re-directs the sampling to other regions of the feasible space by considering previously encountered infeasible samples. The algorithm of GAPSPLIT* with modification (ii) is similar to that described in Section 2.1 with the crucial difference being the iteration step when the solution to the optimisation (7) is infeasible. Here, GAPSPLIT* stores (memorises) the targets that led to the infeasible solutions, and subsequently uses them to avoid sampling at these infeasible targets again. Fig. 2(c) presents a visual illustration of this approach. The set $\Omega^{S'}$ is the set of infeasible samples and $\Omega^{S''} = \Omega^{S'} \cup \Omega^{S}$ is the set of all feasible Ω^{S} and infeasible $\Omega^{S'}$ samples. Fig. 3 shows the algorithmic flowchart of this key difference in GAPSPLIT* modification (ii). If the optimisation is infeasible $\lim_{x_{\tilde{p}} \to T_{\tilde{p}}} g(x) > 0$, then GAPSPLIT* assigns the value of this infeasible primary target $T_{\tilde{p}}$ to the primary variable, and the minimal values x_p^{LB} to all other secondary variables Ω^{Q} of the infeasible sample

$$\begin{aligned} x_{\tilde{p}}^* &= T_{\tilde{p}} \\ x_p^* &= x_p^{LB} \quad \forall\ p \in \Omega^{Q}. \end{aligned} \tag{9}$$

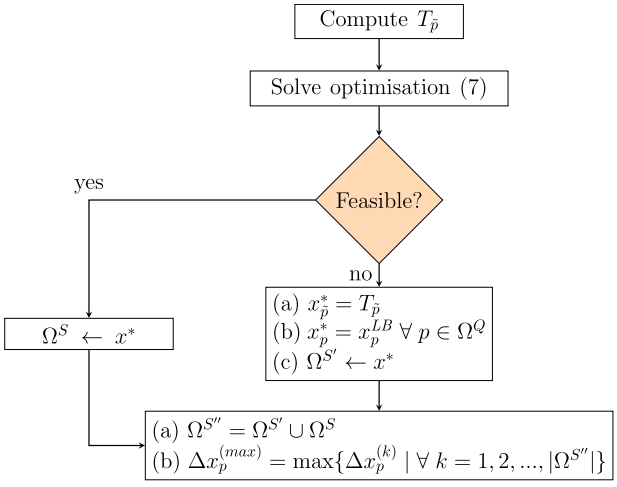


Fig. 3. The iteration step of the proposed modification (ii) of GAPSPLIT* that introduces the set of infeasible samples $\Omega^{S'}$.

Subsequently, GAPSPLIT* adds this infeasible sample x^* to the set of infeasible samples $\Omega^{S'} \leftarrow x^*$. This step implicitly stores the information that the primary target $T_{\tilde{p}}$ is in the infeasible region and allows GAPSPLIT* to disregard the corresponding gap $\Delta x_{\tilde{p}}^{(max)}$ between $x_{\tilde{p}}^{(d)}$ and $x_{\tilde{p}}^{(d-1)}$ in subsequent iterations, and therefore avoids converging to that infeasible sample. Finally, GAPSPLIT* computes the next maximal gaps

$$\Delta x_p^{(max)} = \max\{\Delta x_p^{(k)} \mid \forall\ k = 1, 2, \ldots, |\Omega^{S''}|\} \tag{10}$$

by using feasible and infeasible samples $\Omega^{S''}$, and continues with Eq. (4) and subsequent steps in Section 2.1.

2.3.2. Efficient sorting

The second proposed modification of GAPSPLIT* approach is the efficient sorting of gaps to identify the largest gap in each iteration. This modification addresses the second issue of scalability of the GAPSPLIT

approach in high-dimensional settings as in the power system. The algorithm starts with an initialisation step, followed by iterations that terminate when a stopping criterion is satisfied.

Initially, GAPSPLIT* assigns a set Ω^G to maintain an ordered set of all gaps across all variables which has the cardinality $|\Omega^G| = |\Omega^P| \times |\Omega^S|$. This ordered set contains all gaps $\Delta x_p^{(u)}$ from Eq. (2) for all variables p. The set is

$$\Omega^G = \{\Delta x^{(k)} \mid \forall\, k = 1, 2, \dots, |\Omega^P| \times |\Omega^S|,\ \Delta x^{(k)} \geq \Delta x^{(k-1)}\}, \tag{11}$$

where $\Delta x^{(k)}$ is the kth largest gap across all variables and all samples. The notation of the gap $\Delta x^{(k)}$ drops the index for the sample u and for the variable p for simplicity reasons. The sample index u and the variable index p can be retrieved with the two mappings $\mathcal{U}(k)$ and $\mathcal{P}(k)$, respectively. The overall largest gap is the last element of the ordered set $\Delta x^{(|\Omega^G|)}$ which avoids using the max operators in Eqs. (3)–(4).

In each iteration, GAPSPLIT* locates the primary target $T_{\tilde{p}}$ at the centre of this overall largest gap $\Delta x^{(|\Omega^G|)}$. GAPSPLIT* obtains $d = \mathcal{U}(|\Omega^G|)$, $\tilde{p} = \mathcal{P}(|\Omega^G|)$, and the primary target

$$T_{\tilde{p}} = \frac{x_{\tilde{p}}^{(d+1)} - x_{\tilde{p}}^{(d)}}{2} + x_{\tilde{p}}^{(d)}, \tag{12}$$

where the samples $x_{\tilde{p}}^{(d+1)}$ and $x_{\tilde{p}}^{(d)}$ form the gap $\Delta x^{(|\Omega^G|)}$. Subsequently, GAPSPLIT* selects the secondary variables, for instance with Eq. (6), and then solves optimisation (7) to obtain the optimised OC, the new sample x^*. Subsequently, GAPSPLIT* copies this generated sample x^* in total $|\Omega^P|$ times and inserts one copy each into the sets

$$\Omega_p^{V-S} \leftarrow x^* \qquad \forall\, p \in \Omega^P \tag{13}$$

using the bisection method, which can only be used as the sets Ω_p^{V-S} are ordered. The position of the insertions in the corresponding sets is the map $\mathcal{I}(p)$ such that $x_p^{(\mathcal{I}(p)-1)} \leq x_p^* \leq x_p^{(\mathcal{I}(p)+1)}$. Note that this bisection insertion step is the key advancement as it replaces the sorting step required in each iteration of GAPSPLIT. The reader may recall that the sorting step is the key bottleneck of GAPSPLIT as per analysis in Table 2. However, the bisection method requires only a computational time of $\mathcal{O}(|\Omega^P| \log |\Omega^S|)$ in the worst case. Hence, this efficient bisection step with $\mathcal{O}(|\Omega^P| \log |\Omega^S|)$ replaces the inefficient sorting step with $\mathcal{O}(|\Omega^P||\Omega^S| \log |\Omega^S|)$. Following the insertion, GAPSPLIT* generates $2|\Omega^P|$ new gaps at

$$\begin{aligned} \Delta x_p^{(a)} &= x_p^{(\mathcal{I}(p)+1)} - x_p^* \\ \Delta x_p^{(b)} &= x_p^* - x_p^{(\mathcal{I}(p)-1)}. \end{aligned} \tag{14}$$

Subsequently, GAPSPLIT* inserts these $2|\Omega^P|$ new gaps in $\Omega^G \leftarrow \Delta x_p^{(a)}$, $\Omega^G \leftarrow \Delta x_p^{(b)} \forall\ p \in \Omega^P$ by using the bisection method, as well. GAPSPLIT* limits the cardinality of the set $|\Omega^G| \leq \rho$ to avoid memory issues when the size of $|\Omega^S| \rightarrow a$, where $a \gg 1$ is a large number. In response to this threshold, if $|\Omega^G| > \rho$, then GAPSPLIT* drops the smallest $2|\Omega^P|$ gaps in each iteration

$$\Omega^G \setminus \Delta x^{(k)} \mid \forall\, k = 1, 2, \dots, 2|\Omega^P|, \tag{15}$$

such that $|\Omega^G| \leq \rho$ is satisfied at all times. GAPSPLIT* terminates when sufficient samples are created $|\Omega^S| \geq S$.

3. Measuring performance of samplers

Generic sampling focuses on covering the feasible space with the generated samples Ω^S. A performance metric of such samplers should quantify the coverage of feasible space, which also is a metric for the quality of samples in Ω^S. Such a metric for coverage can also serve as a criterion to stop sampling when the feasible space is sufficiently sampled.

The $\mathcal{COV}$ metric measures the performance of the GAPSPLIT sampler in [45]

$$\mathcal{COV} = 1 - -\frac{1}{|\Omega^P|} \sum_{p=1}^{|\Omega^P|} \frac{\Delta x_p^{(max)}}{r_p} \tag{16}$$

representing the average relative maximal gap $\Delta x_p^{(max)}$ in $|\Omega^P|$ dimensions. To illustrate this metric, the $\mathcal{COV}$ metric has a minimal value $\mathcal{COV} = 0$ when all samples Ω^S are stacked on top of each other, and a maximal value $\mathcal{COV} = 1$ when an infinite number of samples Ω^S are uniformly distributed. For example, $\mathcal{COV} = 0.75$ indicates that the relative maximum gap is 25% on average over all variables Ω^P.

The drawback of analysing sample distributions using the $\mathcal{COV}$ metric is that the analysis focuses on the marginal (univariate) and not the multivariate distribution. Hence, using $\mathcal{COV}$ as a performance metric to assess samplers in high-dimensional settings may result in a poor characterisation of multivariate sample distributions, which is important when using an optimisation procedure to generate the samples (as we will demonstrate in the case study). The example in Fig. 4 illustrates this drawback of using $\mathcal{COV}$ to measure coverage. The samples in the two figures, Figs. 4(a) and 4(b), are clearly differently distributed but have the same marginal distributions in both dimensions. However, as the $\mathcal{COV}$ metric only assesses the marginal univariate distribution, it calculates the same $\mathcal{COV}$ values for the two figures, thereby ignoring the difference in the two bivariate distributions. Hence, the $\mathcal{COV}$ metric is an unsuitable measure of the coverage of samples. Generally in the literature, point-to-point coverage measures fail to quantify how well samples are distributed relative to one another in high dimensional settings and do not account for the distribution of samples in a region [51,50].

Conversely, assessing the volume occupied by the samples seems to be a suitable approach to measure the performance of multivariate sample distributions. The proposed $\mathcal{CHV}$ metric based on computing an approximation of the convex hull volume can overcome the drawback of the $\mathcal{COV}$ metric. The convex hull of a set of samples Ω^S

$$C = \left\{ \lambda^1 x^{(1)} + \dots + \lambda^n x^{(n)} \,\middle|\, \sum_{u=1}^{n} \lambda^u = 1, x^{(u)} \in \Omega^S, \lambda^u \geq 0 \right\} \tag{17}$$

is the smallest convex set that contains all other samples defined in some $|\Omega^P|$-dimensional space, where $x^{(1)}, x^{(2)}, \dots, x^{(n)}$ are independent samples in some Euclidean space $\mathcal{R}^{|\Omega^P|}$, and λ^u are real numbers. The samples $x^{(1)}, x^{(2)}, \dots, x^{(n)}$ are the vertices of the convex hull as they enclose all other samples. The index n represents the number of samples that form the vertices of the convex hull, where $n \leq \Omega^S$. In this work, we use the Qhull algorithm [56] to compute the $\mathcal{CHV}$ metric that measures the convex hull volume occupied by the generated samples Ω^S.

Subsequently, the volume of the convex hull with vertices $x^{(1)}, x^{(2)}, \dots, x^{(n)}$ is

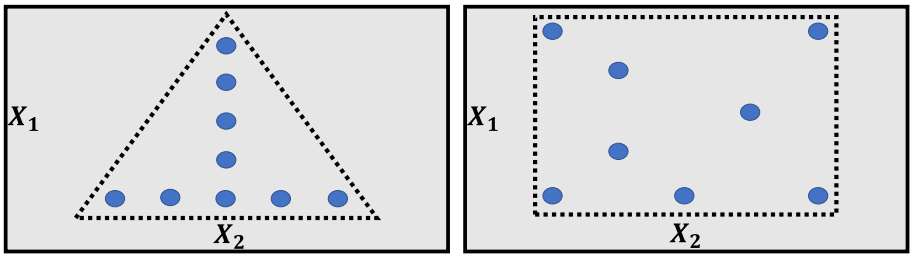
$$\mathcal{CHV} = |\frac{1}{n!} det \begin{pmatrix} x^{(2)} - x^{(1)} & x^{(3)} - x^{(1)} & \cdots & x^{(n)} - x^{(1)} \end{pmatrix}| \tag{18}$$

which further resolves to

$$|\frac{1}{n!} det \begin{pmatrix} x^{(1)} & x^{(2)} & \cdots & x^{(n)} \\ 1 & 1 & \cdots & 1 \end{pmatrix}| \tag{19}$$

However, computing the $\mathcal{CHV}$ in higher dimensions ($|\Omega^P| > 6$) is intractable, as it is a P-hard problem. In this work, we circumvent this issue by randomly selecting a subset of the input variables $\hat{\Omega} \subset \Omega^P; |\hat{\Omega}| \leq 6$ to compute the volume. This random selection allows us to approximate the $\mathcal{CHV}$ in higher dimensions, as we will show in the case study section. The notation $\mathcal{CHV}_{|\Omega^P|=|\hat{\Omega}|}$ denotes the $|\hat{\Omega}|-$dimensional convex hull volume for a set of samples. The reader may refer to the text in [57,58] for further information on convex hulls and their associated volumes.

Finally, we consider other state-of-art coverage metrics in [49]. Specifically, the coefficient of variation between all samples $x_i, x_j \in \mathfrak{R}^{|\Omega^P|}$ is $\lambda = (N \frac{\sum_{i=1}^{N} \gamma_i^2}{(\sum_{i=1}^{N} \gamma_i)^2})^{1/2}$, where $\gamma_i = \min_{i \neq j} |x_i - x_j|$. The smaller the value of λ, the more uniform the distribution of samples and $\lambda = 0$ signifies a perfect uniform mesh. We also consider the point norm distribution, $h = \max_{i=1,2\dots,N} h_i$, where $h_i = \max_{y \in V_i} |x_i - y|$, where h_i is the maximum distance between a sample-point x_i and the points that



(a) $\mathcal{CHV} = 0.5$, $\mathcal{COV} = 0.9$ (b) $\mathcal{CHV} = 1.0$, $\mathcal{COV} = 0.9$

Fig. 4. The samples (blue circles) in (a) and (b) have the same $\mathcal{COV}$ but different $\mathcal{CHV}$ values. The proposed $\mathcal{CHV}$ metric (dotted black line) is suitable for measuring multivariate coverage.

enclose the cell of its Voronoi tessellation V_i. Here also, the smaller the value of h, the more uniform is the distribution. The scalability of λ as the number of samples $|\Omega^S| \rightarrow a$ increase, where $a \gg 1$ is a large number is challenging as it requires $\mathcal{O}(|\Omega^S|^2)$ computations. The scalability of the h coverage measure is similar to that of the proposed $\mathcal{CHV}$ metric. Therefore, λ is not suitable for this application in the paper, however, is included in the comparison for the sake of completeness.

4. Case study

In this section, firstly, we investigate the suitability of the proposed GAPSPLIT* approach to generate representative power systems OCs in comparison to $\mathcal{RS}$ and minimised generation cost $\mathcal{MGC}$ approaches. Secondly, we investigate the performance of the proposed GAPSPLIT* approach to address the low coverage issue of GAPSPLIT when generating OCs for power systems. Thirdly, we show the suitability of the proposed $\mathcal{CHV}$ metric to measure coverage of samples generated by the proposed GAPSPLIT* approach. Fourthly and finally, we discuss the scalability of the proposed $\mathcal{CHV}$ metric to higher dimensions and the computational time of the proposed GAPSPLIT* approach on the IEEE 118-bus system.

4.1. Test system and assumption

The case studies consider the IEEE 6-bus [59] and IEEE 68-bus [60] test systems. Subsequently, using the IEEE 118-bus system [61], we present a scalability study that considers a DC approximation of the power flow. To generate the load profiles, we sample the active loads from a multivariate Gaussian distribution (via Monte Carlo sampling) and assume the correlation between loads to follow Pearson's correlation with a correlation coefficient of 0.75. The distribution was then converted to a marginal Kumaraswamy(1.6, 2.8) distribution using inverse transformation. The reactive loads at the buses scale linearly with active loads by a factor of 0.15 ($\frac{Q}{P} \approx 0.15$). To create the generator profiles, $x_p^* \ \forall \ p \in \Omega^P$, different sampling approaches, including the proposed split-based sampling, $\mathcal{MGC}$, and $\mathcal{RS}$ approaches attempt to solve an optimisation problem that balances generator output with randomly generated loads. The proposed split-based sampling follows the sampling procedure described in Section 2.3, whereas the $\mathcal{MGC}$ approach solves the optimal power flow of the AC-model. Finally, the $\mathcal{RS}$ approach involves sampling generator profiles using a Latin Hypercube Sampling (LHS) procedure and accepting either the LHS generated profile x_p^*, or a perturbation $x_p^* + \delta_{x_p}$, where $\delta_{x_p} \ \forall \ p \in \Omega^P$ are slack variables in the optimisation. The AC models of the networks are used to ensure feasible OCs representing the steady-state operation of the system under AC assumptions. $|\Omega^S| = 1000$ OCs were generated for each sampling approach. After this pre-fault OC data was generated, their corresponding post-fault security labels were simulated with time-domain dynamic simulations. For the simulations, the initial conditions included the pre-fault variables for active and reactive power generations, and active and reactive power loads. The dynamic simulation considered a three-phase fault on line 31–38 for the IEEE-68 bus system with a clearance time of 0.5 s. Subsequently, the simulations were analysed and the post-fault transient security label was computed. The label of an OC was either secure $Y_{i,k} = 0$ when all phase angle differences between any two generators were less than 180° within the 10 s simulation time after the fault, otherwise, the OC was insecure $Y_{i,k} = 1$. To see the generation of database in the context of the final use case, ML models were trained using the generated data. There, the pre-fault OCs and post-fault security labels were used as training databases for quantifying the performance of the trained ML models on testing data. Different ML models, including feed-forward Artificial Neural Network (ANN), Support Vector Machine (SVM), boosting algorithms (Xgboost and Adaboost), and Decision Trees (DTs), were trained as example ML models. The ANNs had three hidden layers with 60, 30, and 10 neurons, respectively, and were trained with a stochastic gradient descent optimiser using the package PyTorch 1.10.0 [62]. The SVM training used a linear kernel, and the boosting algorithms had 50 estimators using the package *scikit-learn* 0.18.1 [63]. DTs were trained with the CART algorithm [64] from the package *scikit-learn* 0.18.1 [63] in Python 3.5.2. The default training settings were selected except using gini impurity instead of entropy to measure the quality of the splits. The data-set was split into training/testing sets in ratio of 75%/25%. 5-fold cross-validation was applied to address under-/overfitting. Subsequently, the Platt method was used to calibrate the score-output S of the classifier [65]. The ML models were evaluated with metrics as the testing accuracy $= \frac{Tp+Tn}{Tp+Tn+Fp+Fn}$, precision $= \frac{Tp}{Tp+Fp}$, specificity $= \frac{Tn}{Tn+Fp}$, and $F1$-score $= \frac{2*precision*specificity}{precision+specificity}$, where Tp and Tn are correctly classified positive and negative OCs, and Fp and Fn are incorrectly classified negative and positive OCs. Additionally, the fraction of insecure OCs $\frac{N_-}{N_-+N_+}$ was computed, where N_+ and N_- are the number of insecure and secure OCs, respectively.

The non-linear optimisation problems were implemented using the package Pyomo 5.6.8 [66] in Python 3.7.4 and solved using IPOPT 3.13.2 [67]. All studies except the scalability section were carried out on a Dell XPS 13 9360 running an Intel(R) Core(TM) i5-8250U processor with 8 GB installed RAM. The scalability study was carried out on a Windows Server 2008 R2 Enterprise running an Intel(R) processor with 96 GB installed RAM. The dynamic simulations are implemented in Julia 1.6.4 with the packages *PowerSystems.jl* [68], *PowerSimulationsDynamics.jl* [69]. The simulations were solved with the IDA package from Sundials solvers [70]. All dynamic simulations were performed on a standard machine with six cores and 16 GB RAM.

4.2. Effective sampling with GAPSPLIT*

In this study, we contrast the performance of the candidate approaches (the proposed GAPSPLIT*, $\mathcal{MGC}$, and $\mathcal{RS}$) in generating representative power systems OCs, which results appear in Fig. 5. Concretely, the figure depicts the 3D-$\mathcal{CHV}$ covered by 5000 OCs generated with the candidate approaches in the IEEE 6- and 68-bus systems.

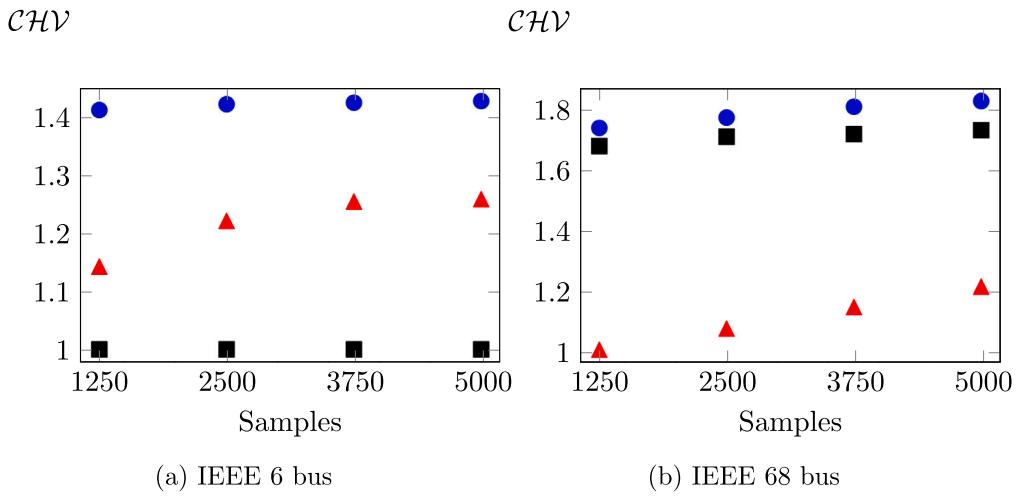


Fig. 5. (a) The $\mathcal{CHV}$ of samples generated using the proposed modification (i) (■), the proposed modification (ii) (●), and $\mathcal{RS}$ (▲).

Table 3
$\mathcal{CHV}$ of 5000 samples computed for 10′000 random variable subsets $\{\hat{\Omega} \subset \Omega^P\}$, where $|\hat{\Omega}| = 3$ for different sampling approaches.

Approach	$\mathcal{CHV}$	
	IEEE 6 bus	IEEE 68 bus
GAPSPLIT*	0.77 ± 0.60	407 ± 474
$\mathcal{MQC}$	0.05 ± 0.24	$< 10^{-11} (< 10^{-11})$
$\mathcal{RS}$	0.70 ± 0.55	296 ± 250

The $\mathcal{CHV}$ values in the figure are normalised with the minimum value for each test system such that $\hat{\mathcal{CHV}} = \frac{\mathcal{CHV}}{min(\mathcal{CHV})}$. In the proposed GAPSPLIT* approach, we consider $|\Omega^{Q^*}| = 0.3|\Omega^P|$ secondary variables.

As evidenced by Fig. 5, the proposed GAPSPLIT* cover a higher 3D-$\mathcal{CHV}$ than the $\mathcal{RS}$ approach, as much as 40% and 55% more in the IEEE 6- and 68-bus systems, respectively. In contrast, as evidenced by Table 3, the proposed GAPSPLIT* approach cover a significantly higher 3D-$\mathcal{CHV}$ than the $\mathcal{MQC}$ approach, in the order of 15× and more than 10^{11}× magnitude, respectively, in the IEEE 6- and 68-bus systems. Admittedly, the poor performance of the $\mathcal{MQC}$ approach is a reflection of its objective function in solving the optimisation problem. Thus, the samples generated by the $\mathcal{MQC}$ approach will only cover a small volume even as the approach generates more OCs, as a result of choosing the same cheap generator combinations to minimise cost.

For a more exhaustive evaluation, we investigate the performance of the proposed GAPSPLIT* and $\mathcal{RS}$ approaches considering 10′000 random variable selections $\{\hat{\Omega} \subset \Omega^P\}$, where $|\hat{\Omega}| = 3$. The results are summarised in Table 3, which shows the $\mathcal{CHV}_{|\Omega^P|=3}$ of the candidate approaches. Overall, the proposed GAPSPLIT* approach cover 10% and 37.5% more volume than the $\mathcal{RS}$ approach in the IEEE 6- and 68-bus systems, respectively.

These results imply that the proposed GAPSPLIT* is suitable for generating a wide range of OCs, which is necessary to enrich the database, especially as the integration of intermittent renewable energy sources becomes the norm.

4.3. Addressing GAPSPLIT issues

In this study, we investigate the performance of the first proposed modification of the GAPSPLIT* approach (Section 2.3.1) to address the low coverage issue of GAPSPLIT (Algorithm 1). We contrast the proposed modification (ii) that introduces the set of infeasible samples $\Omega^{S'}$, the proposed modification (i) that utilises only secondary variables Ω^{Q^*}, and regular GAPSPLIT.

To preface this comparison, regular GAPSPLIT can generate on average three and six unique OCs in the IEEE 6-bus and IEEE 68-bus systems, respectively, before converging to an infeasible region. Subsequently, the maximal gap $\Delta x_{\tilde{p}}^{(max)}$ (Eq. (4)) remains the same, and the algorithm is unable to generate any more feasible OCs. The comparison with the proposed modification (ii) is demonstrated by the results in Fig. 6. Concretely, the figure shows the share of infeasible OCs $\beta = \frac{|\Omega^{S'}|}{|\Omega^{S''}|}$ in the IEEE 6- and 68-bus systems as the candidate sampling approaches generate many OCs $|\Omega^S| \to a$, where $a \gg 1$ is a large number. As evidenced by Fig. 6, the proposed modification (ii) has a higher value of β in earlier iterations for both systems that decrease as more OCs are generated. The value of β decreases from 19.5% when $|\Omega^{S''}| = 210$ to 4.9% when $|\Omega^{S''}| = 15593$ in the IEEE 6-bus system, and from 29.3% when $|\Omega^{S''}| = 3622$ to 21.3% when $|\Omega^{S''}| = 9366$ in the IEEE 68-bus system. This downward trend of β indicates an improved performance of the proposed modification (ii) as the algorithm generates more OCs. The proposed modification (ii) works for both small and relatively large systems as β decreases when $|\Omega^S|$ grows in both systems. In contrast, the share of infeasible samples β increases in both systems for regular GAPSPLIT. On the other hand, from this perspective of β, the proposed modification (i) has the best performance as it generates on average only one infeasible OC in both systems. Modification (ii) also avoids converging to infeasible regions. Admittedly, the modification (i) trivially avoids converging to infeasible regions as its optimisation discards the hard constraint on the primary target. Additionally, its objective function aims to minimise the distance to the candidate targets. However, in terms of generating OCs in a non-convex and disconnected feasible space, as is the case in power systems, these results indicate that the first proposed modification of GAPSPLIT* improves on the low coverage issue of GAPSPLIT.

Furthermore, we note that modification (ii) is preferred to modification (i) in small systems by the results in Figs. 7(b) and 7(a). The figures show a scatter-plot of OCs generated by the modifications (ii) and (i), respectively. As evidenced by the figures in Fig. 7, the OCs generated with modification (ii) cover the entire feasible space and not only the boundaries, and is thus the preferred approach. However, this preference of modification (ii) over modification (i) is not entirely visible in larger systems. In larger systems (e.g., IEEE 68-bus), there is a higher share of infeasible OCs β (e.g., β =5.1% and 23.5%, respectively, in the IEEE 6- and 68-bus systems). This higher value of β in the IEEE 68-bus system denotes an increase in the number of iterations required by the algorithm before termination, and invariably, an increase in the computation time of modification (ii). On that note of computation time, modification (i) is suitable for large networks. However, the $\mathcal{CHV}$ comparison between the two approaches in Fig. 5(b) indicates

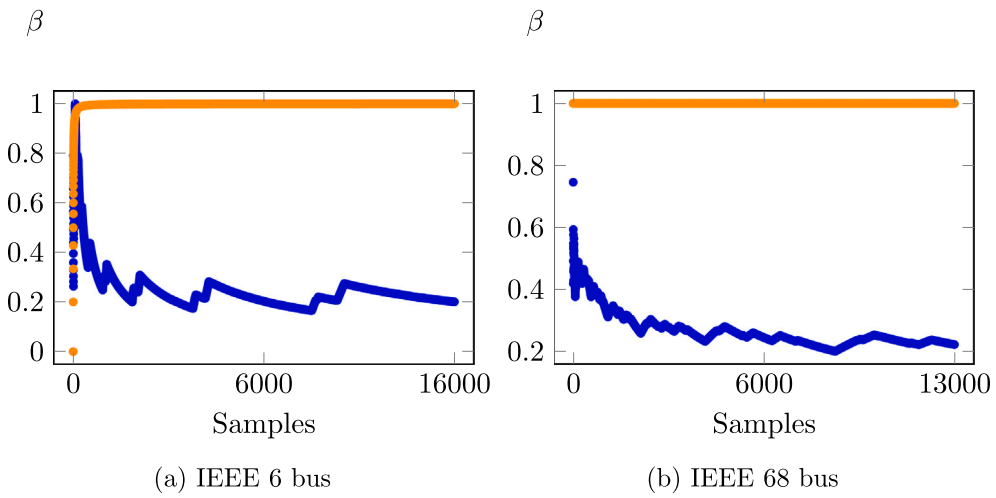


Fig. 6. The value of $\beta = \frac{|\Omega^{s'}|}{|\Omega^{s''}|}$ for the proposed modification (ii) (●) reduces while regular GAPSPLIT's (●) increases as more samples are generated in both (a) small and (b) larger systems.

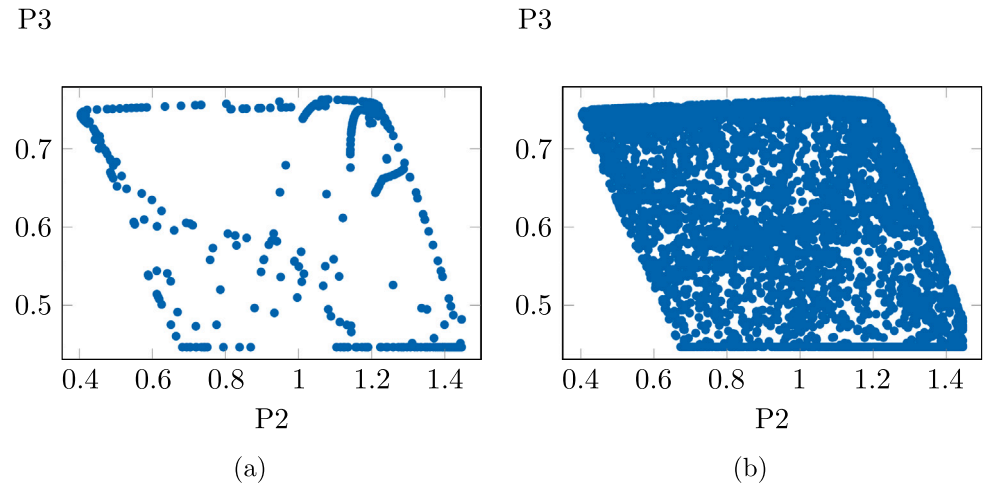


Fig. 7. Modification (ii) better distributes samples across the feasible than modification (i) in small systems, respectively in (b) and (a).

that coverage of OCs generated using the proposed modification (ii) is marginally better than modification (i). In the rest of the manuscript, unless otherwise stated, we consider the proposed modification (ii) as GAPSPLIT*.

It is also worth highlighting that GAPSPLIT* sampling with both primary and secondary variables is preferred over GAPSPLIT* sampling with only primary variables, as demonstrated by the distribution of samples in Figs. 9(b)–9(a).

4.4. Measuring performance of samplers

This case study contrasts the proposed $\mathcal{CHV}$ metric with the $\mathcal{COV}$ metric to measure coverage of the feasible space by generated OCs. As an illustrative example on the IEEE 68-bus system, we use the two candidate metrics to compute coverage of different multivariate distributions in Figs. 9(a) and 9(b). Concretely, the figures depict a scatter-plot of OCs generated using GAPSPLIT* sampling with only primary variables and GAPSPLIT* sampling with both primary and secondary variables, respectively.

As evidenced by Fig. 9, the $\mathcal{COV}$ value is the same in both Figs. 9(a) and 9(b), while the $\mathcal{CHV}$ value is approximately 100% higher in Fig. 9(b) than in Fig. 9(a). This result shows that the $\mathcal{COV}$ does not distinguish between different multivariate distributions.

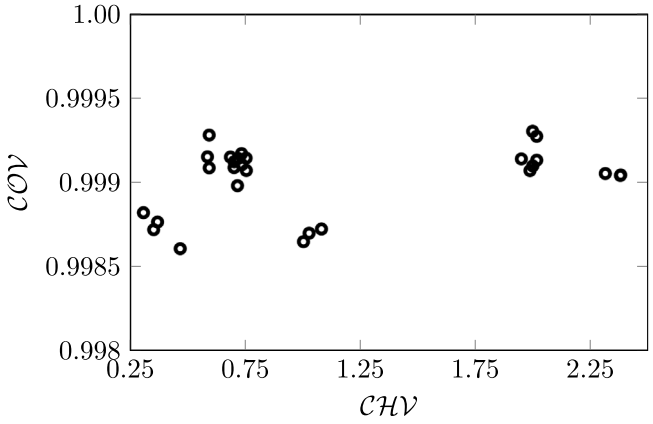
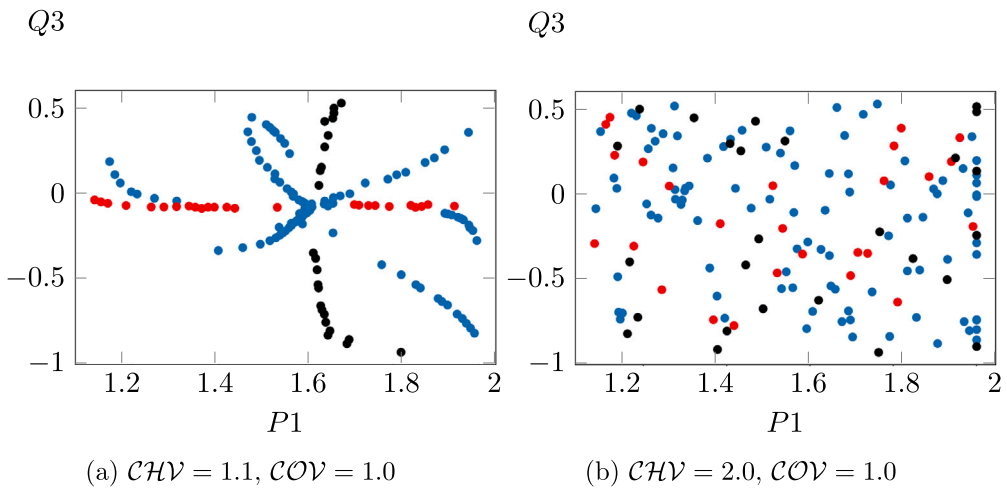


Fig. 8. Samples that have similar $\mathcal{COV}$ values are differentiated with the proposed $\mathcal{CHV}$.

For a more exhaustive evaluation, we consider four different sets of 5000 OCs that are generated by varying the number of secondary variables $|\Omega^{Q*}| = \{0, 1, 2, 3\}$ in the GAPSPLIT* algorithm. Subsequently, we compute the $\mathcal{COV}$ and $\mathcal{CHV}$ of the different sets considering 10′000



(a) $\mathcal{CHV} = 1.1$, $\mathcal{COV} = 1.0$ (b) $\mathcal{CHV} = 2.0$, $\mathcal{COV} = 1.0$

Fig. 9. (a) GAPSPLIT* with only primary targets (shown for $P1$ (●) and $Q3$ (●)) generates samples that are distributed along the axis of that variable and do not cover the entire feasible space. (b) Proposed use of GAPSPLIT* uses primary and secondary targets to cover the entire feasible space.

Table 4
The $\mathcal{CHV}$, $\mathcal{COV}$, h and λ values of four different sets of 5000 samples computed for 10′000 random variable selections $\{\hat{\Omega} \subset \Omega^P\}$, where $|\hat{\Omega}| = 3$.

$\lvert\Omega^{Q*}\rvert$	$\mathcal{CHV}$	$\mathcal{COV}$	h	λ
0	0.49 ± 0.36	0.99(< 0.01)	9.84 ± 0.16	0.75 ± 0.15
1	0.99 ± 0.78	0.99(< 0.01)	9.75 ± 0.24	0.60 ± 0.06
2	1.02 ± 0.72	0.99(< 0.01)	9.90 ± 0.07	0.46 ± 0.04
3	1.09 ± 0.86	0.99(< 0.01)	9.83 ± 0.18	0.49 ± 0.04

Table 5
Computation time to generate 100'000 samples in IEEE 118-bus generator space.

Approach	Average time	Total time
GAPSPLIT*	(0.57 ± 0.20) s	16 h
GAPSPLIT	(0.78 ± 0.24) s	21 h
$\mathcal{MGC}$	(0.60 ± 0.20) s	17 h
$\mathcal{RS}$	(0.64 ± 0.28) s	17 h

Table 6
ANN trained for dynamic security on 1000 OCs from IEEE 68-bus system.

Approach	$\frac{N_+}{N_-+N_+}$	$F1$-score	Accuracy	Precision	Specificity
GAPSPLIT*	93.3%	99.5%	98.4%	99.9%	99.9%
$\mathcal{RS}$	84.4%	93.7%	91.2%	93.5%	90.0%

random variable selections $\{\hat{\Omega} \subset \Omega^P\}$, where $|\hat{\Omega}| = 3$. The results are in Fig. 8, which depicts a scatter-plot of the proposed $\mathcal{CHV}$ metric against the $\mathcal{COV}$ metric for the same sets of OCs. Overall, the proposed $\mathcal{CHV}$ can distinguish the coverage of 'good' from 'bad' sample distributions while $\mathcal{COV}$ cannot. Concretely, the proposed $\mathcal{CHV}$ has a wider range of values $(0.25, 2.50)$ and higher standard deviation of 0.74 than the $\mathcal{COV}$ metric with values ranging between $(0.9985, 0.9995)$ and standard deviation of < 0.001. Additionally, other coverage metrics like λ and h range between $(0.65, 1.02)$ and $(9.89, 9.99)$, respectively, with standard deviations of 0.18 and 0.03 for the same dataset. There, just as $\mathcal{COV}$, h cannot distinguish 'good' from 'bad' sample distributions, while λ can distinguish. However, as we pointed out in Section 3, the metric λ is not suitable as it does not computationally scale well to large number of samples. The $\mathcal{CHV}$ metric considers the multivariate distribution of OCs to measure coverage, and that makes it a better metric to quantify the spread of OCs across multidimensional space. Table 4 summarises the comparison between $\mathcal{COV}$, λ, h and the proposed $\mathcal{CHV}$ metrics for the different sets of sample distributions, showing that the proposed $\mathcal{CHV}$ is more suitable to measure coverage by differentiating the distinct sets of OCs.

4.5. Computational performance & scalability

This case study tests the computational performance and scalability of the proposed split-based sampling approach and the proposed coverage metric to larger systems. The performance was tested for the number of OCs generated and the size of the power system (number of dimensions of variables). In this study, GAPSPLIT was modified to store infeasible OCs for comparison (to prevent early convergence to an infeasible OC), while the proposed GAPSPLIT* approach is modified as described in Section 2.3.2. To study the scalability of the $\mathcal{CHV}$ metric, on the IEEE 118-bus system, 100 random subsets of variables with dimension sizes $|\Omega^P| = \{2-7\}$ and sample size $|\Omega^S| = 5000$ are drawn and the $\mathcal{CHV}$ is computed for each subset of variables. Fig. 10(a) shows that the random selection of variables does not influence the $\mathcal{CHV}$ and the mean and median values of the normalised $\mathcal{CHV}$ are suitable to approximate the $\mathcal{CHV}$ for dimension sizes $|\Omega^P| = \{2-7\}$. Fig. 10(b) shows the relationship between the $\mathcal{CHV}$ of random subsets for dimensions 3 and 7. The correlation shows that the average value of $\mathcal{CHV}_{|\Omega^P|=3}$ is sufficient to approximate $\mathcal{CHV}_{|\Omega^P|=7}$, and therefore computing a reduced $\mathcal{CHV}$ is a good estimator for $\mathcal{CHV}$ in higher dimensions.

Table 5 shows the computational times to generate 100′000 OCs with different approaches on the IEEE 118-bus system. GAPSPLIT takes 21 hours, in contrast to 17 hours by the $\mathcal{MGC}$ and $\mathcal{RS}$ approaches. Albeit, the OCs generated using GAPSPLIT cover a 30% larger volume than OCs generated using $\mathcal{RS}$. This increase in total time for GAPSPLIT is as a result of increased time to sort Ω^S and find the maximal gap $\Delta x_{\tilde{p}}^{(max)}$ as more OCs are generated. The moving average (with a sliding window of 1000) of the time it takes to sample an OC for GAPSPLIT shows a linear increase over time, with a slope angle $\angle\frac{\Delta t}{\Delta s}$ of 45°. The proposed second modification of the GAPSPLIT* approach from Section 2.3.2 mitigates this increase in time by regulating the size of the set of gaps and efficiently sorting newly generated OCs.

4.6. Dynamic security and machine learning

This case study tests the generated data when applied to the intended use case of ML-based DSA on the IEEE 68-bus system. The dynamic security labels of OCs from GAPSPLIT* and $\mathcal{RS}$ were simulated, and different ML models including SVM, Adaboost, Xgboost, DT, and ANN were trained. The results in Table 6 show that the generated data from GAPSPLIT* results in better performances when training an ANN across the metrics of test accuracy, $F1$-score, precision, and

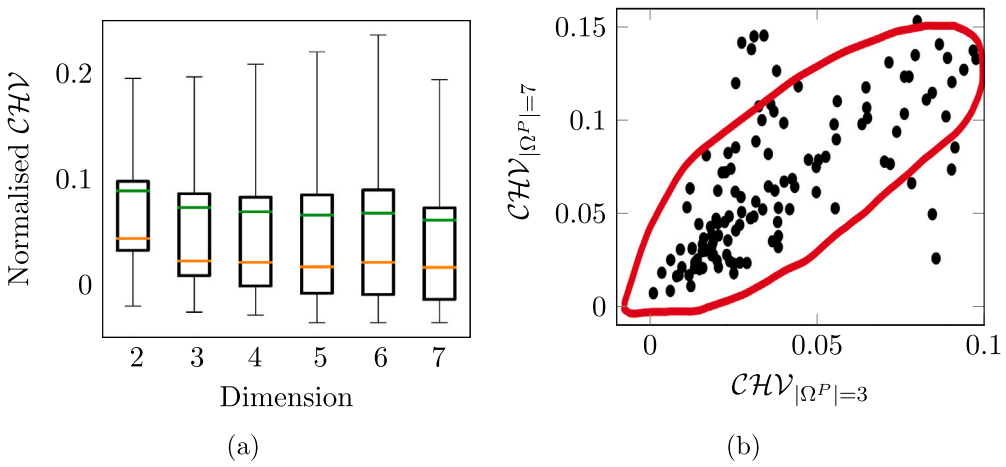


Fig. 10. (a) The normalised $\mathcal{CHV}$ mean (—) and median (—) values of random subset selections are similar across different dimensions (b) There is some correlation between the $\mathcal{CHV}$ of a random subset and the $\mathcal{CHV}$ of the full dimension.

Table 7
DT trained for dynamic security on 1000 OCs from IEEE 68-bus system.

Approach	$\frac{N_+}{N_-+N_+}$	$F1$-score	Accuracy	Precision	Specificity
GAPSPLIT*	93.3%	99.6%	99.2%	99.6%	99.6%
$\mathcal{RS}$	84.4%	99.2%	98.8%	99.0%	99.5%

Table 8
$F1$-score for 5 different ML models trained on 1000 OCs from the IEEE 68-bus system. Each type of model is trained 100 times.

Approach	SVM	Adaboost	Xgboost	DT	ANN
GAPSPLIT*	99.0%	99.5%	99.2%	99.5%	98.8%
$\mathcal{RS}$	96.0%	98.0%	98.6%	98.2%	91.9%

specificity by 7.2%, 5.8%, 6.4% and 9.9%, respectively. GAPSPLIT* generated more insecure OCs which can enhance the prediction accuracy of predicting insecure OCs. Maximising the accuracy for insecure OCs and reducing false negatives is important as these type of errors can lead to power blackouts which are significantly worse than false positives. For DTs the values remained similar in GAPSPLIT* and $\mathcal{RS}$ as shown in Table 7. For a more exhaustive comparison, each of SVM, Adaboost, Xgboost, DT, and ANN models were trained 100 times on data from GAPSPLIT* and $\mathcal{RS}$. The results in Table 8 show that the generated data results in marginally better performance across the two approaches for SVM, Adaboost, Xgboost, and DT models. A 6.9% improvement is recorded for the ANN.

5. Conclusion

A systematic approach to creating representative databases is pivotal to the adoption of ML methods for real-time (dynamic-) security assessment. This work proposes a novel split-based sampling approach GAPSPLIT* to generate representative samples that systematically explore the feasible space of power systems. The key feature of the split-based sampling is the ability to consider model-based constraints $g(x) \leq 0$ when generating a sample (OC) in the optimisation. When using this sampling approach for power systems, the physical constraints can be considered for the steady-state in the form of the AC network power flow constraints, as used when optimising the generator dispatches in an ACOPF model. The proposed split-based sampling aims for diverse data by jumping from one part of the solution space to another underrepresented part to cover a larger space (distribution) with fewer OCs. In the IEEE 68-bus system, samples generated using the proposed split-based sampling cover 37.5% more volume than with $\mathcal{RS}$. The proposed $\mathcal{CHV}$ is better suited than distance-based metrics to quantify the performance of a *generic sampler* and differentiate good from bad sample distributions. The proposed split-based approach takes 0.57 s on average to generate samples for the IEEE 118-bus system. Future work will involve exploiting historical data in the sampling procedure to generate new OCs that improve the information gain of the classifier. There, the proposed algorithm as a sequential process shall consider another variable that creates balanced datasets. Our vision is to use this proposed algorithm as a baseline then consider "active learning" that can use discriminative information on the class distribution in the sequential sampling process.

CRediT authorship contribution statement

Al-Amin B. Bugaje: Conceptualization, Methodology, Data curation, Writing – original draft, Visualization, Investigation, Formal analysis, Software, Validation. **Jochen L. Cremer:** Conceptualization, Methodology, Supervision, Writing – review & editing. **Goran Strbac:** Writing – review & editing, Project administration, Funding acquisition, Resources.

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.

Data availability

Data will be made available on request.

Acknowledgements

This work was supported by a scholarship funded by the Nigerian National Petroleum Corporation, NG, the TU Delft AI Labs Programme, NL, and the research project IDLES, UK (EP/R045518/1).

References

[1] Panciatici P, Bareux G, Wehenkel L. Operating in the fog: Security management under uncertainty. IEEE Power Energy Mag 2012;10:40–9. http://dx.doi.org/10.1109/MPE.2012.2205318.
[2] Cui M, Wang J, Tan J, Florita AR, Zhang Y. A novel event detection method using PMU data with high precision. IEEE Trans Power Syst 2019;34:454–66. http://dx.doi.org/10.1109/TPWRS.2018.2859323.
[3] Zhao J, Netto M, Huang Z, Yu S, Gomez-Exposito A, Wang S, et al. Roles of dynamic state estimation in power system modeling, monitoring and operation. IEEE Trans Power Syst 2020;1. http://dx.doi.org/10.1109/TPWRS.2020.3028047.

[4] Konstantelos I, Jamgotchian G, Tindemans SH, Duchesne P, Cole S, Merckx C, et al. Implementation of a massively parallel dynamic security assessment platform for large-scale grids. IEEE Trans Smart Grid 2017;8:1417–26. http://dx.doi.org/10.1109/TSG.2016.2606888.

[5] Duchesne L, Karangelos E, Wehenkel L. Recent developments in machine learning for energy systems reliability management. Proc IEEE 2020;108:1656–76. http://dx.doi.org/10.1109/JPROC.2020.2988715.

[6] He M, Zhang J, Vittal V. Robust online dynamic security assessment using adaptive ensemble decision-tree learning. IEEE Trans Power Syst 2013;28:4089–98.

[7] Zhu L, Lu C, Dong ZY, Hong C. Imbalance learning machine-based power system short-term voltage stability assessment. IEEE Trans Ind Inf 2017;13:2533–43.

[8] Wang B, Fang B, Wang Y, Liu H, Liu Y. Power system transient stability assessment based on big data and the core vector machine. IEEE Trans Smart Grid 2016;7:2561–70.

[9] James J, Hill DJ, Lam AY, Gu J, Li VO. Intelligent time-adaptive transient stability assessment system. IEEE Trans Power Syst 2017;33:1049–58.

[10] Zhang Y, Xu Y, Dong ZY, Zhang R. A hierarchical self-adaptive data-analytics method for real-time power system short-term voltage stability assessment. IEEE Trans Ind Inf 2018;15:74–84.

[11] Xu Y, Dai Y, Dong ZY, Zhang R, Meng K. Extreme learning machine-based predictor for real-time frequency stability assessment of electric power systems. Neural Comput Appl 2013;22:501–8.

[12] Wang Q, Li F, Tang Y, Xu Y. Integrating model-driven and data-driven methods for power system frequency stability assessment and control. IEEE Trans Power Syst 2019;34:4557–68.

[13] Zhang T, Sun M, Cremer JL, Zhang N, Strbac G, Kang C. A confidence-aware machine learning framework for dynamic security assessment. IEEE Trans Power Syst 2021.

[14] Bugaje A-AB, Cremer JL, Sun M, Strbac G. Selecting decision trees for power system security assessment. Energy AI 2021;6:100110.

[15] Cremer JL, Strbac G. A machine-learning based probabilistic perspective on dynamic security assessment. Int J Electr Power Energy Syst 2021;128:106571.

[16] Liu Y, Wang J, Yue Z. Improved multi-point estimation method based probabilistic transient stability assessment for power system with wind power. Int J Electr Power Energy Syst 2022;142:108283.

[17] Papadopoulos PN, Milanović JV. Probabilistic framework for transient stability assessment of power systems with high penetration of renewable generation. IEEE Trans Power Syst 2016;32:3078–88.

[18] Bellizio F, Cremer JL, Sun M, Strbac G. A causality based feature selection approach for data-driven dynamic security assessment. Electr Power Syst Res 2021;201:107537.

[19] Liu Y, Shi X-J, Xu Y. A hybrid data-driven method for fast approximation of practical dynamic security region boundary of power systems. Int J Electr Power Energy Syst 2020;117:105658.

[20] Bellizio F, Bugaje A-AB, Cremer JL, Strbac G. Verifying machine learning conclusions for securing low inertia systems. Sustain Energy Grids Netw 2022;30:100656.

[21] Sevilla FRS, Liu Y, Barocio E, Korba P, Andrade M, Bellizio F, et al. State-of-the-art of data collection, analytics, and future needs of transmission utilities worldwide to account for the continuous growth of sensing data. Int J Electr Power Energy Syst 2022;137:107772.

[22] Yan R, Geng G, Jiang Q. Data-driven transient stability boundary generation for online security monitoring. IEEE Trans Power Syst 2020;36:3042–52.

[23] Zhu L, Hill DJ. Data/model jointly driven high-quality case generation for power system dynamic stability assessment. IEEE Trans Ind Inf 2021;18:5055–66.

[24] Stiasny J, Chevalier S, Nellikkath R, Sævarsson B, Chatzivasileiadis S. Closing the loop: A framework for trustworthy machine learning in power systems. 2022, arXiv preprint arXiv:2203.07505.

[25] Hand DJ. Principles of data mining. Drug Saf 2007;30:621–2.

[26] Konstantelos I, Sun M, Tindemans SH, Issad S, Panciatici P, Strbac G. Using vine copulas to generate representative system states for machine learning. IEEE Trans Power Syst 2018;34:225–35.

[27] Sun M, Konstantelos I, Strbac G. A deep learning-based feature extraction framework for system security assessment. IEEE Trans Smart Grid 2018;10:5007–20.

[28] Kroese DP, Taimre T, Botev ZI. Handbook of monte carlo methods, 706. John Wiley & Sons; 2013.

[29] Huang T-e, Guo Q, Sun H. A distributed computing platform supporting power system security knowledge discovery based on online simulation. IEEE Trans Smart Grid 2016;8:1513–24.

[30] Wang G, Guo J, Ma S, Zhang X, Guo Q, Fan S, et al. Data-driven transient stability assessment with sparse PMU sampling and online self-check function. CSEE J Power Energy Syst 2022.

[31] Ren C, Xu Y. A fully data-driven method based on generative adversarial networks for power system dynamic security assessment with missing data. IEEE Trans Power Syst 2019;34:5044–52.

[32] Liu C, Sun K, Rather ZH, Chen Z, Bak CL, Thøgersen P, et al. A systematic approach for dynamic security assessment and the corresponding preventive control scheme based on decision trees. IEEE Trans Power Syst 2014;29:717–30.

[33] Krishnan V, McCalley JD, Henry S, Issad S. Efficient database generation for decision tree based power system security assessment. IEEE Trans Power Syst 2011;26:2319–27.

[34] Hamon C, Perninge M, Söder L. An importance sampling technique for probabilistic security assessment in power systems with large amounts of wind power. Electr Power Syst Res 2016;131:11–8.

[35] Genc I, Diao R, Vittal V, Kolluri S, Mandal S. Decision tree-based preventive and corrective control applications for dynamic security enhancement in power systems. IEEE Trans Power Syst 2010;25:1611–9.

[36] Zhu L, Hill DJ, Lu C. Semi-supervised ensemble learning framework for accelerating power system transient stability knowledge base generation. IEEE Trans Power Syst 2021;37:2441–54.

[37] Jafarzadeh S, Genc VMI. Probabilistic dynamic security assessment of large power systems using machine learning algorithms. Turk J Electr Eng Comput Sci 2018;26:1479–90.

[38] Wu Q, Koo TJ, Susuki Y. Dynamic security analysis of power systems by a sampling-based algorithm. ACM Trans Cyber-Phys Syst 2018;2:1–26.

[39] Thams F, Venzke A, Eriksson R, Chatzivasileiadis S. Efficient database generation for data-driven security assessment of power systems. IEEE Trans Power Syst 2020;35:30–41. http://dx.doi.org/10.1109/TPWRS.2018.2890769.

[40] Venzke A, Molzahn DK, Chatzivasileiadis S. Efficient creation of datasets for data-driven power system applications. Electr Power Syst Res 2021;190:106614.

[41] Joswig-Jones T, Baker K, Zamzam AS. 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. IEEE; 2022, p. 1–5.

[42] Kaufman DE, Smith RL. Direction choice for accelerated convergence in hit-and-run sampling. Oper Res 1998;46:84–95.

[43] Haraldsdóttir HS, Cousins B, Thiele I, Fleming RM, Vempala S. CHRR: Coordinate hit-and-run with rounding for uniform sampling of constraint-based models. Bioinformatics 2017;33:1741–3.

[44] Saa PA, Nielsen LK. Ll-ACHRB: A scalable algorithm for sampling the feasible solution space of metabolic networks. Bioinformatics 2016;32:2330–7.

[45] Keaty TC, Jensen PA. Gapsplit: Efficient random sampling for non-convex constraint-based models. Bioinformatics 2020;36:2623–5.

[46] Jenior ML, Moutinho Jr TJ, Dougherty BV, Papin JA. Transcriptome-guided parsimonious flux analysis improves predictions with metabolic networks in complex environments. PLoS Comput Biol 2020;16.

[47] Nadal IV, Chevalier S. Optimization-based exploration of the feasible power flow space for rapid data collection. 2022, arXiv preprint arXiv:2206.12214.

[48] Mukherjee R, De A. Real-time dynamic security analysis of power systems using strategic PMU measurements and decision tree classification. Electr Eng 2021;103:813–24.

[49] Gunzburger M, Burkardt J. Uniformity measures for point sample in hypercubes. Rapp tech, Florida State University (Cf P 73); 2004.

[50] Aggarwal CC, Hinneburg A, Keim DA. On the surprising behavior of distance metrics in high dimensional space. In: International conference on database theory. Springer; 2001, p. 420–34.

[51] Domingos P. A few useful things to know about machine learning. Commun ACM 2012;55:78–87.

[52] Zimek A, Schubert E, Kriegel H-P. A survey on unsupervised outlier detection in high-dimensional numerical data. Stat Anal Data Min: ASA Data Sci J 2012;5:363–87.

[53] Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: Synthetic minority over-sampling technique. J Artificial Intelligence Res 2002;16:321–57.

[54] He H, Bai Y, Garcia EA, Li S. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In: 2008 IEEE international joint conference on neural networks. IEEE; 2008, p. 1322–8.

[55] Molzahn DK, Hiskens IA. A survey of relaxations and approximations of the power flow equations. Now Publishers; 2019.

[56] Barber CB, Dobkin DP, Huhdanpaa H. The quickhull algorithm for convex hulls. ACM Trans Math Software 1996;22:469–83.

[57] Boyd S, Boyd SP, Vandenberghe L. Convex optimization. Cambridge University Press; 2004.

[58] Stein P. A note on the volume of a simplex. Amer Math Monthly 1966;73:299–301, URL: http://www.jstor.org/stable/2315353.

[59] Wood AJ, Wollenberg BF, Sheblé GB. Power generation, operation, and control. John Wiley & Sons; 2013.

[60] Pal B, Chaudhuri B. Robust control in power systems. Springer Science & Business Media; 2006.

[61] Illinois institute of technology (IIT), IEEE 118-bus system data. 2013, URL: http://motor.ece.iit.edu/Data/.

[62] Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, et al. Pytorch: An imperative style, high-performance deep learning library. Adv Neural Inf Process Syst 2019;32:8026–37.

[63] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine learning in Python. J Mach Learn Res 2011;12:2825–30.

[64] Breiman L, Friedman JH, Olshen RA, Stone CJ. Classification and regression trees, 432. International Group; 1984, p. 151–66.
[65] Platt J. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. Adv Large Margin Classif 1999;10:61–74.
[66] Hart WE, Laird CD, Watson J-P, Woodruff DL, Hackebeil GA, Nicholson BL, et al. Pyomo-optimization modeling in python, 67. Springer; 2017.
[67] Wächter A, Biegler LT. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math Program 2006;106:25–57.
[68] Lara JD, Barrows C, Thom D, Krishnamurthy D, Callaway D. PowerSystems. jl—A power system data management package for large scale modeling. SoftwareX 2021;15:100747.
[69] Henriquez-Auba R, Lara JD, Callaway DS, Barrows C. Transient simulations with a large penetration of converter-interfaced generation: Scientific computing challenges and opportunities. IEEE Electrif Mag 2021;9:72–82. http://dx.doi.org/10.1109/MELE.2021.3070939.
[70] Hindmarsh AC, Brown PN, Grant KE, Lee SL, Serban R, Shumaker DE, et al. SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers. ACM Trans Math Softw 2005;31:363–96.