




# Data-Driven Power System Operation: Exploring the Balance Between Cost and Risk

Jochen L. Cremer , *Graduate Student Member, IEEE*, Ioannis Konstantelos , *Member, IEEE*,  
Simon H. Tindemans , *Member, IEEE*, and Goran Strbac, *Member, IEEE*

**Abstract**—Supervised machine learning has been successfully used in the past to infer a system’s security boundary by training classifiers (also referred to as security rules) on a large number of simulated operating conditions. Although significant research has been carried out on using classifiers for the detection of critical operating points, using classifiers for the subsequent identification of suitable preventive/corrective control actions remains underdeveloped. This paper focuses on addressing the challenges that arise when utilizing security rules for control purposes. Illustrative examples and case studies are used to show how even very accurate security rules can lead to prohibitively high risk exposure when used to identify optimal control actions. Subsequently, the inherent tradeoff between operating cost and security risk is explored in detail. To optimally navigate this tradeoff, a novel approach is proposed that uses an ensemble learning method (AdaBoost) to infer a probabilistic description of a system’s security boundary. Bias in predictions is compensated by the Platt Calibration method. Subsequently, a general-purpose framework for building probabilistic and disjunctive security rules of a system’s secure operating domain is developed that can be embedded within classic operation formulations. Through case studies on the IEEE 39-bus system, it is showcased how security rules derived from supervised learning can be efficiently utilized to optimally operate the system under multiple uncertainties while respecting a user-defined balance between cost and risk. This is a fundamental step toward embedding data-driven models within classic optimisation approaches.

**Index Terms**—Supervised machine learning, AdaBoost, power systems operation, security rules, dynamic stability.

## I. INTRODUCTION

THE increasing complexity of power systems as well as the growing uncertainty that surrounds operation, introduced by renewable sources of energy and changing demand patterns, has rendered critical the use of advanced operation tools for ensuring system stability [1], also known as operational reliability

Manuscript received April 16, 2018; revised July 23, 2018; accepted August 19, 2018. Date of publication August 27, 2018; date of current version December 19, 2018. This work was supported by a studentship funded by the Engineering and Physical Sciences Research Council. We are thankful to Nicolas Omont and colleagues from Réseau de Transport d’Électricité who provided expertise that greatly assisted the research. Paper no. TPWRS-00571-2018. (*Corresponding author: Jochen L. Cremer.*)

J. L. Cremer, I. Konstantelos, and G. Strbac are with the Department of Electrical and Electronic Engineering, Imperial College London, London SW7 2AZ, U.K. (e-mail: j.cremer16@imperial.ac.uk; i.konstantelos@imperial.ac.uk; g.strbac@imperial.ac.uk).

S. H. Tindemans is with the Department of Electrical Sustainable Energy, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: s.h.tindemans@tudelft.nl).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TPWRS.2018.2867209

[2]. Under this new reality, a new breed of security assessment approaches has emerged, combining data-driven statistical inference and machine learning within a Monte Carlo framework.

### A. Existing Approaches

In general, data-driven work-flows follow three main steps: (i) Generate a population of possible operating points that may arise in the next hours/days by sampling from statistical models fitted to past historical data. (ii) For each sampled operating point, perform a simulation for each credible contingency scenario and determine post-fault security. (iii) Using the system’s pre-fault state variables as features and the post-fault security status as a label, construct classifiers (also known as security rules) using standard machine learning algorithms such as Decision Trees (DTs). The principal idea is that a Transmission System Operator (TSO) or Distribution System Operator (DSO) can carry out the above training procedure in a periodic and offline manner and construct classifiers that can be used as predictors to infer the post-fault security status of unseen operating points. Subsequently, at each control period, the TSO or DSO can generate a very large number of possible operating points and rapidly classify them as safe or unsafe without performing time-consuming simulations. Such an analysis can identify critical operating points that could lead to security problems, providing insight to operators and flagging them up for further analysis.

In general, the aim of such work-flows is to provide a scalable way of managing uncertainty and system complexity within the tight constraints of real-time operation in an effort to improve the TSO’s and DSO’s situational awareness. As such, most research efforts until now have focused on studying the computational performance of such platforms [3] as well exploring the statistical model for generating operating points in a multivariate setting (e.g. [4], [5]) and machine learning approaches for building useful security rules [1].

Currently, the security rules are primarily used to identify problematic operating points that may arise in the near future. The next natural step is to use these security rules to determine what kind of control actions should be performed by the TSO and DSO to bring the system back to the secure domain. In other words, instead of limiting the use of security rules to *classification* purposes, it is possible to use security rules as a guide for steering the system back to a safe operating domain using a suitable *control* framework, as illustrated in Fig. 1. Much less work has been carried out on this latter topic. We begin by presenting the two control approaches that have been investigated

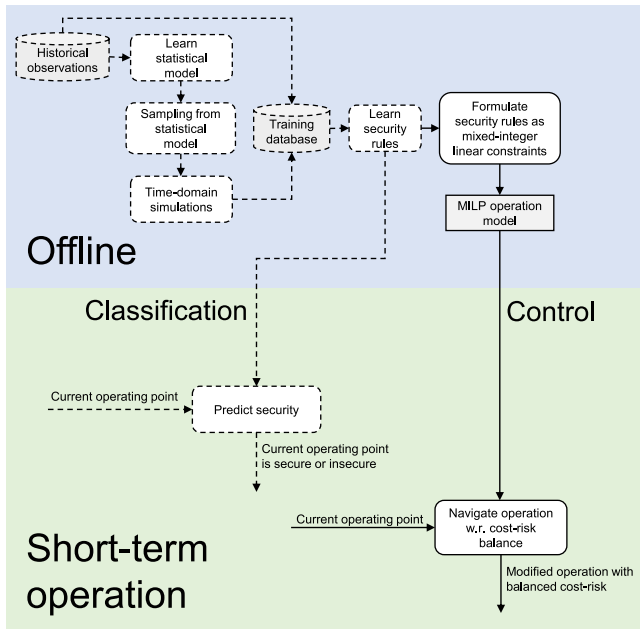


Fig. 1. Data-driven work-flows for classification (dashed lines) and the proposed control purpose (straight lines).

in the literature and then highlight the open questions we aim to address.

The first type of approaches used in the past is based on a heuristic analysis of the obtained security rules, so as to derive re-dispatch rules for preventive or corrective control. For example, authors in [6], [7] construct security rules in the form of Decision Trees (DTs). Following DT training, an operating point that is classified as unsafe can be brought back to safety by changing the variables present in the parent nodes of the DT node in question (each DT node is associated with a particular feature and threshold value).

The second type makes use of optimization, where security rules are embedded within an Optimal Power Flow (OPF) problem. The idea is that security rules partition the pre-fault operating space into regions of post-fault security and regions of post-fault insecurity. The aim is to ensure that all operating points that may arise within the next hour (or another adopted control time frame) can be guided towards one of the safe regions using some preventive/corrective control action. In the case of DTs, these constraints can be included in the OPF formulation in the form of inequality expressions. For example, in [8], [9]) one OPF problem is formulated for each secure terminal node of the trained DT. All problems are solved and the solution resulting in the smallest operating cost can be adopted as the most cost-effective way to ensure post-fault security. Authors in [10] take a different approach, where post-fault security is ensured by conservatively adjusting the bounds of generators found to be potentially lead to insecurity. The most cost-effective trajectory is identified by constraining operation within a decision surface that respects power balance and all other relevant scheduling constraints. Instead of adjusting generator bounds, line flow limits are considered as features in [11], where the authors propose a Mixed Integer Linear Programming (MILP) approach to embed

the entire DT in a single problem. Also an MILP is used in [12] and an offline-learned *single safety margin* is used to deal with potential insecurity. Although such approaches are promising and a natural step towards fully automated and comprehensive control frameworks under uncertainty, they face several challenges.

### B. Challenges of Data-Driven Operation

The first challenge refers to the fact that a classifier's accuracy when applied to a classification task can be radically different to the same classifier's accuracy when applied to a control task. This is because the population of operating points used to train a security rule is fundamentally different to the population of operating points that arise as a result of an optimal control process.

As explained in detail later, although this may appear to be a subtle point, it is crucial since there can be cases where a 99.9% accurate security rule (i.e. extremely good in identifying critical points) results in 0% accuracy when used to derive optimal control actions (i.e. the system is erroneously guided to an unsafe region believed to be safe). The implications of this issue, which has not been studied in detail in the existing literature, can be problematic.

The second challenge has to do with the fact that since the trained classifiers are by definition imperfect, this inadvertently raises the issue of managing the risk that arises while also being cost-optimal i.e. tackling the risk-cost balance. The impact of imperfections in security classifiers has been investigated in isolation, for example in [13]. To deal with the risk of imperfect classifiers researchers have proposed several methods to learn risk-averse security rules. For example, [10] and [11] propose to asymmetrically adjust the weights (*asym. weighting*) of safe and unsafe operating points during training. By increasing the weight of insecure training samples, the boundary is approximated more conservatively. However, apart from several other drawbacks, such conservative approaches can have a detrimental effect on operating cost. Other authors introduce bias after training. For example, [6] and [7] verify the validity of identified control actions by executing simulations; this procedure is repeated until a certificate of security can be obtained for the new operating point. However, such approaches require a large number of simulations in the control period, resulting in a prohibitively large computational load and cost inefficiencies. To avoid cost inefficiencies, researchers have been balancing risk and cost in non-data-driven approaches; e.g. [14] employs a particle-swarm optimization, [15] a multi-objective optimization, and [16] a chance-constrained and multi-objective (stochastic) optimization. However, the challenge to describe and balance the risk in data-driven approaches caused by the imperfection of the classifiers remains unaddressed.

The third challenge has to do with the applicability of security rules to unseen operating conditions. In the past, researchers have developed heuristics that are able to improve performance when dealing with unseen operating conditions. However such methods entail large realtime computational load since they require knowledge of the specific operating point so as to modify

the base case control scheme accordingly (e.g., [6], [7], [10]). In this paper we investigate generalizable ways to improve control scheme robustness.

### C. Present Work

In this paper, two approaches are proposed to address the aforementioned challenges in different ways. The first approach addresses the first two aforementioned challenges by showing how to learn operation safety margins so as to conservatively approximate the region of safe operation subject to a user-specified tolerance. Instead of generalizing a *single safety margin* across all security rules, as has been done in the past [10], [12], *condition-specific safety margins* are tailored to each individual condition of the security rules, resulting in cost savings; consequently the approach improves the risk/cost balance indirectly by learning those *condition-specific safety margins*.

The second approach proposes a novel *risk-averse* methodology to address all three aforementioned challenges. The concept is to balance the pre-fault operating cost and expected probability of operating within an acceptable region via a multi-objective optimization framework. This entails a fundamental shift from deterministic to probabilistic treatment of security which is enabled by moving from the use of DTs, which have traditionally been used in the past, to ensemble methods such as AdaBoost [17]. Starting with uniformly weighted training samples, AdaBoost can iteratively train base DT estimators by adapting the sample weights in each iteration. The final ensemble (consisting of all trained base estimators) can be used to provide probability estimates regarding the post-fault security of a particular operation region based on the individual votes of the base estimators. We also show how those estimates must be calibrated using Platt Calibration [18] to deal with the bias typically introduced by boosting algorithms. The bias is reduced by fitting a sigmoid function to the probability estimates and the posterior probabilities. To embed the security rules in the optimization problem, both approaches involve Generalized Disjunctive Programming (GDP) [19]. GDP uses binary and continuous variables to exploit the inherent logic structure of the security rules in order to reduce the combinatorics. The formulation of GDP enables solvers to make use of branch-and-bound search in order to achieve superior computational performance. We show how the developed methods result in computationally efficient approaches, rendering them suitable for real-time deployment in large systems.

To study the proposed approaches, an IEEE 39-bus case study is used. First, we show that existing approaches, primarily focused at training classifiers for predicting safe/unsafe labels for unseen operating points, are inherently ill-suited for the task of identifying suitable control actions. We proceed by showing that both proposed approaches are able to drive system operation much closer to the global optimum than existing approaches, while also abiding to the user-defined risk tolerance level. Moreover, we show that the proposed *risk-averse* approach is capable of identifying cost-effective control actions under a large range of unseen operation conditions.

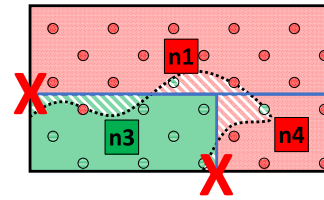


Fig. 2. Pre-fault feature space in  $\mathbb{R}^2$ : the true boundary (for  $\mathcal{Y}$ , dotted black) is estimated by a DT using acceptable (green circles) and unacceptable (red circles) training samples; the estimated boundary (for  $\hat{\mathcal{Y}}$ , blue line) divides the space into acceptable (green) and unacceptable (red) regions. Wrong estimations (shaded) can be critical (red X).

The rest of the paper is structured as follows. In Section II, we present in detail the challenges of inferring suitable control actions on the basis of data-driven proxies of security. Thereafter, in Section III, the approach to learn *condition-specific safety margins* is introduced. Subsequently, the *risk-averse* approach is proposed in Section IV and the case study is presented in Section V. Finally, Section VI is the conclusion.

## II. DATA-DRIVEN SECURITY RULES

### A. Security Rules for Classification

We first consider supervised classification methods that can predict the security of an operating point. For such a task, the usual approach is to use a binary class label (acceptable or unacceptable) corresponding to the post-fault state of the system subject to a user-specified binary criterion (e.g. line overloads, over-voltages, transient stability etc.). To train and assess the performance of a classifier, two data sets are usually distinguished: the training data  $(X, Y)$  and the test data  $(X^t, Y^t)$ . The population of pre-fault operating points  $X$  and  $X^t$  can, for example, be obtained by sampling an underlying statistical model fitted to historical data, while the population of labels  $Y$  and  $Y^t$  is obtained via simulation  $\mathcal{Y} : X \mapsto Y$  (see [3], [20] and [21] for details). A classifier is trained on data  $(X, Y)$  containing  $\mathcal{N}$  samples  $(x^i, y^i), i = 1, \dots, \mathcal{N}$  of operating points, where  $x^i \in \mathbb{R}^p$  is a vector of  $p$  features (pre-fault state variables, such as line flows, power of generators and loads) and  $y^i \in \{0, 1\}$  is the corresponding class label with  $y^i = 1$  and  $y^i = 0$  signifying acceptable and unacceptable post-fault operation, respectively. In this paper we focus on training binary DTs using the Classification And Regression Trees (CART) algorithm [22]. A typical DT, as illustrated in Fig. 2 for  $p = 2$ , divides the entire pre-fault operating space in regions of unacceptable (red) and acceptable (green) post-fault behaviour with class label  $\{0, 1\}$ . Each region corresponds to a terminal node  $n \in \Omega_T$  that are associated with one of the class labels  $\{0, 1\}$  denoted by  $\Omega_T^0$  and  $\Omega_T^1$ , respectively. This notation corresponds to  $\Omega_T^0 = \{n1, n4\}$  and  $\Omega_T^1 = \{n3\}$  in the Fig. 2. This association is determined based on the fraction of training points  $(X, Y)$  in each terminal node that have the class label  $\{0, 1\}$ . This fraction also provides a probability estimate of the prediction  $\hat{\mathcal{P}}^0(x^t), \hat{\mathcal{P}}^1(x^t)$ . Consequently, the prediction of an unseen operating point  $x^t$  is



obtained by the predominating probability estimate, such as

$$\hat{Y}(x^t) = \begin{cases} 0 & \text{if } \hat{\mathcal{P}}^0(x^t) > \hat{\mathcal{P}}^1(x^t) \\ 1 & \text{if } \hat{\mathcal{P}}^0(x^t) \leq \hat{\mathcal{P}}^1(x^t). \end{cases} \quad (1)$$

However, due to the limiting nature of the DT (i.e. linear conditions) and/or insufficient training, the predicted class  $\hat{y}^t$  may be wrong  $\hat{y}^t \neq y^t$  (since  $\hat{Y}$  is an approximation of  $\mathcal{Y}$ ). For example, the DT in Fig. 2 is approximating the true boundary (dotted black line) which is non-linear and thus cannot be perfectly inferred. This is evidenced by the fact that terminal node n3 is not pure, but contains mixed class labels. Furthermore, wrong predictions may occur when the DT has been trained on an insufficiently large number of samples, or if the training and testing populations differ [3], e.g. due to the respective underlying model to generate the samples, such as in a topological change in the power system (e.g., as studied in [13]). Such misclassifications are unavoidable when constructing a classifier, and for this reason quantifying the quality of the classifier is important.

One typical measure of a classifier's quality is the test error rate (e.g., used in [8], [10], [11]), denoted  $\zeta$ . The test error rate is calculated based on data  $(X^t, Y^t)$  containing  $\mathcal{N}^t$  samples that were unseen in the training procedure  $X^t \cap X = \emptyset$  and the population of predicted class labels  $\hat{Y}^t$  (obtained from  $\hat{Y} : X^t \mapsto \hat{Y}^t$ ), such that  $\zeta = \frac{|Y^t \neq \hat{Y}^t|}{\mathcal{N}^t}$ , where  $|\cdot|$  denotes cardinality. However, in this paper we show that although metrics such as  $\zeta$  can be useful in quantifying classification performance, they cannot predict a rule's performance when used for inferring suitable mitigation control actions.

### B. Security Rules for Control

As mentioned in the introduction, the natural step after obtaining a set of security rules is to develop an optimization framework that identifies control actions so that the system is contained within one of the prescribed safe operating regions while achieving minimum operating cost. This can be formulated as the following optimization problem:

$$\begin{aligned} \min_{x^*} \quad & f(x^*) \\ \text{s.t.} \quad & h(x^*) = 0 \\ & g(x^*) \leq 0 \\ & q_{(x,y)}(x^*) \leq 0, \end{aligned} \quad (2)$$

where  $f(x^*)$  is the operation cost and  $g(x^*)$  and  $h(x^*)$  denote the inequality and equality constraints of the power system respectively and  $x^*$  is the vector of operational decision variables such as generator injections, line flows etc. The embedded security rules are denoted as  $q_{(x,y)}$  and bound variables  $x^*$  in the regions of acceptable operation.

Due to the non-perfect nature of  $q_{(x,y)}$ , there will be cases where optimization problem (2) drives operation to regions of the feature space that turn out to be unacceptable i.e. the operation vector  $x^*$  is classified as safe according to the security rules (i.e.  $\hat{Y}(x^*) = 1$ ), but found to be unsafe when the contingency is simulated (i.e.  $\mathcal{Y}(x^*) = 0$ ). As such, when analysing a

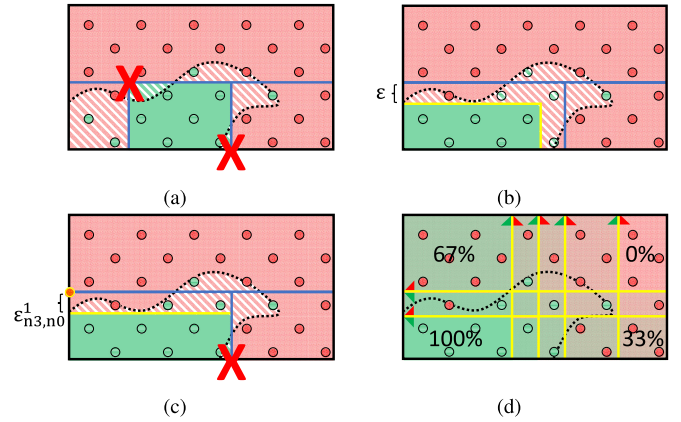


Fig. 3. Different approaches to obtain data-driven security rules for control (based on Fig. 2). (a) *Asym.-weighting*. (b) *Single- $\epsilon$* . (c) *Condition-specific- $\epsilon$* . (d) *Risk-averse*. The approach-specific modifications are shown as yellow lines.

population of  $\mathcal{N}^*$  optimised operating points  $X^*$  by predicting  $\hat{Y} : X^* \mapsto \hat{Y}^*$  and computing the true labels  $\mathcal{Y} : X^* \mapsto Y^*$ , the control error  $k \in [0, 1]$  can be expressed as the ratio of incorrectly classified points  $k = \frac{|\hat{Y}^* \neq Y^*|}{\mathcal{N}^*}$ .

Note that control error  $k$  is very different to the classification error  $\zeta$  since it refers to a fundamentally different population of operating points. Whereas the population  $X^t$  used to compute  $\zeta$  is drawn from the same distribution as the training data  $X$ , the population  $X^*$  that determines  $k$  results from an optimization procedure that favors cost-minimizing regions (see [8], [9], [11]), where the security rules endogenously restrict the problem's feasible region according to  $q_{(x,y)}$ . As a result, if the optimization is linear, then  $X^*$  accumulates upon the binding hyperplanes, since, according to the fundamental theorem of linear programming [23], the optimal solution always lies either on one of the vertices of the feasible region or on a connecting line of two optimal vertices. This is especially critical if we consider the fact that unacceptable operating regions can be less costly than acceptable regions since the latter may entail some preventive/corrective measures entailing an increase in cost. For example, referring back to Fig. 2, if there is a low-cost point that has been included in the set of acceptable terminal nodes (denoted by red X marks) then it is possible to obtain a control error  $k \gg \zeta$  and as high as 1. One intuitive approach to address this problem is the *asymmetric weighting* approach to conservatively approximate the boundary with the idea of shifting the binding hyperplanes towards the acceptable region (as done in [10], [11]). However, this shift is not straightforward to control and still results in a control error  $k \gg \zeta$ , as illustrated in Fig 3(a) (denoted by the red X marks).

In response, we investigate two strategies to achieve low  $k$ :

- 1) Under-estimate the acceptable operation regions by introducing some safety margin  $\epsilon$ .
- 2) Provide an explicit term in the objective function of optimization problem (2) so as to penalize operation in regions with non-zero  $\zeta$ .

We present the first *condition-specific- $\epsilon$*  approach in Section III and the second *risk-averse* approach in Section IV.

### III. COMPUTING CONDITION-SPECIFIC SAFETY MARGINS

Modifying security rules with a safety margin in order to increase control robustness has been proposed in the past [10], [12]. Nevertheless, the existing literature has exclusively focused on cases using a safety margin in an iterative online search (e.g., [10]) dealing with a single contingency and largely ignored the economic implications of introducing such a margin. In this paper, in an effort to develop a scalable data-driven framework, we focus on cases involving multiple contingencies and computing margins outside the control time frame. In such a case, two approaches can be adopted to reduce  $k$ . The first approach is to apply a *single safety margin*  $\varepsilon$  to all conditions of all rules, as shown in Fig. 3(b). As can be seen, this approach can lead to the unnecessary shrinkage of the estimated acceptable operating region, thus potentially leading to increased cost. The second approach is to compute a *condition-specific- $\varepsilon$*  for each individual condition of each DT rule, as shown in Fig. 3(c). Inaccurate conditions are identified, and a safety margin  $\varepsilon$  is iteratively added so as to shift the estimated boundary towards the actual acceptable region. This shift is biased by the set of conditions that are identified as shortcomings of the classifier and get improved. With respect to this bias, the complete elimination of inaccuracies of conditions cannot be guaranteed, as exemplified by the red X in Fig. 3(c).

#### A. Mathematical Formulation

In this section we build upon optimization problem (2). We adopt the standard DC Optimal Power Flow (OPF) formulation and modify it so as to include security rules with a safety margin. GDP [19] is used to transform the DT to a set of inequality constraints  $q_{(x,y)}(x)$  that can be embedded in the optimization. The logic is that each terminal node of the DT labelled as acceptable  $n \in \Omega_T^1$  corresponds to the disjunction of all parent branching nodes. To formulate such a disjunction, a convex-hull reformulation [24] or big- $M$  reformulation [19] can be used. In this application, we adopt the big- $M$  reformulation which results in fewer constraints and variables. The reformulation of the constraints in iteration  $j$  (the safety margin  $\varepsilon$  changes at each iteration) is

$$a_m^\top x \leq (s_m - \varepsilon_{n,m}^j) b_n + a_m^\top M_1 (1 - b_n) \quad (3)$$

$\forall n \in \Omega_T^1$  and  $\forall m \in \Omega_A^L(n)$ , where  $\Omega_A^L(n) \in \Omega_B$  are all ancestor branch nodes that provide a left ( $\leq$ ) condition on the path from the initial node  $n_0$  to the terminal node  $n$ . Accordingly,  $\Omega_A^R(n)$  is the set of all ancestor branch nodes providing a right ( $>$ ) condition

$$a_m^\top x > (s_m + \varepsilon_{n,m}^j) b_n + a_m^\top M_2 (1 - b_n) \quad (4)$$

$\forall n \in \Omega_T^1$  and  $\forall m \in \Omega_A^R(n)$ . The original conditions obtained from the DT learning algorithm are the feature threshold  $s_m$  and  $a_m = e_{h(m)}$  for each branch node  $\forall m \in \Omega_B$ , where  $e_{h(m)}$  is the  $h$ th standard basis vector in the  $p$ -dimensional space.  $b_n = \{0, 1\}$  is a binary variable for each of the disjunct  $n \in \Omega_T^1$ ; if  $b_n = 1$ , operation in terminal node  $n$  is selected. Exactly one disjunction must be selected according to  $\sum_{n \in \Omega_T^1} b_n = 1$ . Note that strict inequalities cannot be modelled in optimizations,

therefore a small  $\beta \in \mathbb{R}_{>0}$  can be added to the right-hand-side of Eq. (4).

The big- $M$  constants have vector form  $M_1 \in \mathbb{R}^p$  and  $M_2 \in \mathbb{R}^p$ , where  $p$  are the features. In order to speed-up the computations, it is critical to use small big- $M$  values; large enough to ensure the desired behaviour but not unnecessarily large so as to increase the problem's feasible region.

$$M_1 = \max \left\{ a_m s_m + \bar{a}_m x^L : m \cup_{n \in \Omega_T^1} \Omega_A^L(n) \right\} \quad (5a)$$

$$M_2 = \min \left\{ a_m s_m + \bar{a}_m x^U : m \cup_{n \in \Omega_T^1} \Omega_A^R(n) \right\}, \quad (5b)$$

where max and min are operators to compare element-wise the vector entries,  $\bar{a}_m = 1 - a_m$  is the negation of  $a_m$  and it is assumed that all linear (feature) variables are bounded  $x^L \leq x \leq x^U$ . As illustrated in Fig. 3(c), the safety margins  $\varepsilon_{n,m}^j$  are iteratively increased for the conditions of the branch nodes  $m$  of each rule from initial node  $n_0$  to the terminal node  $n$ . The corresponding safety margins to be increased from iteration  $j$  to  $j + 1$  are identified in an offline search procedure. Initially, all safety margins are  $\varepsilon_{n,m}^0 = 0 \forall n \in \Omega_T^1, \forall m \in \Omega_A^L(n) \cup \Omega_A^R(n)$ . Then, in each  $j$ , the critical conditions  $\Omega_{C,j}$  are searched by taking a test set of optimized operating points ( $X^*$ ), where the dispatch decisions were computed using the proposed optimization accounting for the corresponding  $\varepsilon_{n,m}^j$ . Subsequently, for each optimized operating point  $x^* \in X^*$ , the true class label is computed through the true function (e.g., via simulations)  $y^* = \mathcal{Y}(x^*)$  and the critical conditions  $(n, m)$  are those on which the unacceptable operating points ( $y^* = 0$ ) accumulate. For each unacceptable point  $x^* \in (X^*)$ , the conditions  $(n, m)$  are identified if the following condition  $R(n, m, x^*)$  holds:

$$R = \begin{cases} |a_m^\top x^* - (s_m + \varepsilon_{n,m}^j)| \leq \delta & \text{if } m \in \Omega_A^R(n) \\ |a_m^\top x^* - (s_m - \varepsilon_{n,m}^j)| \leq \delta & \text{if } m \in \Omega_A^L(n), \end{cases} \quad (6)$$

where  $x^*$  is located in terminal node  $n$  and  $\delta$  is a tolerance parameter. If this holds,  $(n, m)$  is added to the set of critical conditions  $\Omega_{C,j} = \Omega_{C,j} + (n, m)$ . After those conditions  $(n, m)$  are identified  $\forall x^* \in (X^*)$ , the corresponding safety margins are increased  $\varepsilon_{n,m}^{j+1} = \varepsilon_{n,m}^j + \Delta\varepsilon \quad \forall (n, m) \in \Omega_{C,j}$  by a user-specified step  $\Delta\varepsilon$ .

### IV. DATA-DRIVEN RISK-AVERSE OPERATION

As discussed previously, the second strategy for achieving a low control error  $k$  is to introduce an explicit term to the objective function (2) which penalizes risk exposure. In this section we achieve this by using DT ensembles. We begin by introducing ensemble learning techniques with a focus on AdaBoost and Platt calibration. We then introduce the risk-averse formulation that enables a user-defined trade-off between operational cost and risk exposure.

#### A. DT Ensembles

Ensembles are classifiers combining the classification output of a set of simple classifiers  $\Omega_L$  into one single classification output [25]. For instance, the diverse outputs of 6 simple classifiers

(DTs), each with two terminal nodes are shown in Fig. 3(d), where each simple classifier's decision boundary is a yellow line. The final output can be obtained as probability estimates (probability estimates for the acceptable class are in the figure) by combining the votes of the simple classifiers. This reduces the risk of wrong classification [25] and has been shown to result in a better approximation  $\hat{\mathcal{Y}}(x)$  of the true function to compute the label  $\mathcal{Y}(x)$ , where  $x$  is the feature vector of the operating point. This better result requires that the individual classifiers are diverse and more accurate than random [26]. Two different concepts exist for computing the predicted label: *Majority* and *Soft Voting*. In Majority Voting, each base estimator  $l \in \Omega_L$  provides a class label  $\hat{\mathcal{Y}}_l(x)$ , while in Soft Voting each  $l$  provides a probability estimate  $\hat{\mathcal{P}}_l^0(x)$  and  $\hat{\mathcal{P}}_l^1(x)$  for each class label  $\{0, 1\}$  [27]. In Soft Voting, the probability estimate that an unseen operating point  $x^t$ , for instance, belongs to the acceptable class is computed as

$$\hat{\mathcal{P}}_E^1(x^t) = \frac{1}{|\Omega_L|} \sum_{l \in \Omega_L} \hat{\mathcal{P}}_l^1(x^t). \quad (7)$$

The predicted class label can be obtained by using those probability estimates in Eq. (1). Overall, many algorithms exist to learn a DT ensemble. In this paper, and after extensive testing not shown here, we choose to use the AdaBoost. In AdaBoost, at each iteration of the training process, the weight of each training sample is adjusted proportionally to the current misclassification error. AdaBoost by default employs Majority Voting. The extension to Soft Voting, called SAMME.R was introduced in [28], where it was shown to outperform other approaches in terms of convergence time and test error. However, since the re-weighting of boosting algorithms biases the probability estimates, calibration is required.

### B. Calibration

Boosting methods, such as AdaBoost, tend to push the predicted probabilities away from 0 and 1, resulting in a distortion in the estimated probabilities [29]. Calibration is used to correct this distortion by mapping the probability estimates to the posterior probabilities. Two methods are typically used for calibration, differing in the mapping function. Isotonic Regression [30] uses a free-form monotonically increasing line, while Platt Calibration [18] uses a sigmoid function. The sigmoid function  $P_E^1$  is fitted using maximum likelihood estimates of a new training dataset  $(X^c, Y^c)$  (e.g., as plotted in Fig. 4). In this paper, Platt calibration is used, since according to the literature it yields the best probability estimates when combined with the AdaBoost algorithm [29]. In addition, as we show below, the sigmoid function can be linearised and embedded within a MILP problem.

### C. Mathematical Formulation

The balance between cost and risk optimization has been widely researched. Here, we show how two approaches could be used to account for the risk of unacceptable operation. Whereas both consider the same specific constraints  $q_{(x,Y)}(x)$ , they differ in the way they account for  $k(x)$ . In the first approach, the

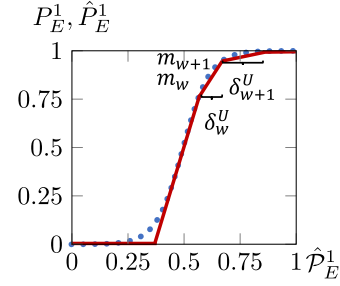


Fig. 4. Calibration of the risk function: Calibrated probability estimate  $P_E^1$  (dotted blue) and linear approximation  $\hat{P}_E^1$  (red). The uncalibrated probability estimate is the  $\hat{P}_E^1$ -axis.

standard OPF formulation is constrained by  $k(x) \leq \gamma$ , where  $\gamma$  is a user-specified parameter to limit  $k(x) \in [0, 1]$ . In this paper, we study and propose the second approach, where  $k(x)$  is accounted for in the objective function,

$$(1 - \alpha)f'(x) + \alpha k(x), \quad (8)$$

of a multi-objective optimization with linear scalarization. The control error  $k(x)$  and the normalized operating cost  $f'(x) \in [0, 1]$  are weighted using the trade-off factor  $\alpha$ . By increasing the parameter  $\alpha \in [0, 1]$ , the user can select more risk-averse operation. To compute the normalized operating cost  $f'(x) \in [0, 1]$ , the standard DCOPF linear cost function is averaged over all generators and scaled to the minimal and maximal generator costs.

To implement  $k(x)$ , the constraints from Section III are modified and new constraints are taken into account. The probability estimated of the base estimators are

$$\hat{\mathcal{P}}_l^1 = \sum_{n \in \Omega_{T,l}} \hat{\mathcal{P}}_{n,l}^1 b_{n,l} \quad \forall l \in \Omega_L, \quad (9)$$

where  $\hat{\mathcal{P}}_{n,l}^1$  is the probability estimate for acceptable operation in the terminal node  $n$  and is obtained by computing the ratio of acceptable training operating points in each terminal node  $n$ . As in Section III, the binary variable  $b_{n,l}$  corresponds to the terminal node  $n$  to be selected for operation ( $b_{n,l} = 1$ ). To extend the disjunctive formulation to a DT ensemble, some modifications are undertaken: all remaining inequality and equality constraints are extended for each base learner  $l \in \Omega_L$  as follows:

$$a_{m,l}^\top x \leq s_{m,l} b_{n,l} + a_{m,l}^\top M_{1,l} (1 - b_{n,l}) \quad (10)$$

$\forall n \in \Omega_{T,l}$  and  $\forall m \in \Omega_{A,l}^L(n)$ . Note, all terminal nodes  $\Omega_{T,l}$  are considered and all parameters, such as  $a_{m,l}$ ,  $s_{m,l}$ ,  $M_{1,l}$ , as well as the sets  $\Omega_{T,l}$ ,  $\Omega_{A,l}^L$  are extended by the index  $l$ . Accordingly, the right branch nodes  $\Omega_{A,l}^R$  are considered in

$$a_{m,l}^\top x \geq s_{m,l} b_{n,l} + \beta + a_{m,l}^\top M_{2,l} (1 - b_{n,l}) \quad (11)$$

$\forall n \in \Omega_{T,l}$  and  $\forall m \in \Omega_{A,l}^R(n)$  with the big- $M$  value  $M_{2,l}$ . Exactly one disjunction must be selected for each  $l \in \Omega_L$  according to  $\sum_{n \in \Omega_{T,l}} b_{n,l} = 1$ . The optimal big- $M$  values  $\forall l \in \Omega_L$



are calculated as follows:

$$M_{1,l} = \max\{a_{m,l}s_{m,l} + \bar{a}_{m,l}x^L : m \cup_{n \in \Omega_{T,l}} \Omega_{A,l}^L(n)\} \quad (12a)$$

$$M_{2,l} = \min\{a_{m,l}s_{m,l} + \bar{a}_{m,l}x^U : m \cup_{n \in \Omega_{T,l}} \Omega_{A,l}^R(n)\} \quad (12b)$$

In order to include the non-linear sigmoid function  $P_E^1$  within our MILP problem, piece-wise linearization is employed to obtain the approximation  $\hat{P}_E^1$ . We approximate this function by using  $|\Omega_W|$  line segments. For  $\mathcal{P}_E^1 \geq 0.5$ , we can avoid introducing a binary variable, since  $\frac{\partial^2 P_E^1}{\partial (\hat{P}_E^1)^2} \leq 0$ . However, to approximate for  $\hat{P}_E^1 < 0.5$ , we introduce one single binary variable  $b' = \{0, 1\}$  to account for an initial line segment  $w = 0$ . The linear approximation  $\hat{P}_E^1$  is illustrated in Fig. 4 and formulated using the following constraints:

$$\hat{P}_E^1 = \sum_{w \in \Omega_W} m_w \delta_w \quad (13a)$$

$$\hat{P}_E^1 = b' \delta_0 + \sum_{w \in \Omega_W \setminus \{0\}} \delta_w \quad (13b)$$

$$0 \leq \delta_0 \leq b' \delta_0^U \quad (13c)$$

$$b' \delta_w^U \leq \delta_w \leq \delta_w^U, \quad \forall w \in \Omega_W \setminus \{0\} \quad (13d)$$

where  $m_w$  is declining ( $m_1 \geq m_2 \geq m_3 \dots$ ) for  $w \geq 1$  and  $m_0 = 0$ . Consequently,  $k(x) = \hat{P}_E^1 = 1 - \hat{P}_E^1$ . Note, in the absence of calibration,  $k(x) = 1 - \hat{P}_E^1$  and Eq. (13) becomes redundant.

## V. CASE STUDY

A number of studies have been undertaken to provide insights in the theory being discussed and to provide evidence for the efficacy of the proposed approaches. After stating the case study assumptions, we show the mismatch in quantifying the rule-quality when used for classification and for computing control actions. Subsequently, we show the performance of the proposed approaches with respect to balancing cost and risk of unacceptable operation and the sensitivity of this balance. We continue by providing the result of a study on the applicability to unseen operating conditions and finish with discussing the scalability of the approaches.

### A. Test System and Assumptions

The IEEE 39 bus system was used: all data was taken from [31] and modified (as in [12], including post-fault redispatching of generator power levels by  $\pm 100$  MW) to ensure N-1 SCOPF feasibility for all samples. The acceptability class label is computed by proving if the energy balance can be maintained after a fault. If this was the case for all line outages, the pre-fault operating point  $x$  was considered as acceptable  $\mathcal{Y}(x) = 1$  and otherwise unacceptable  $\mathcal{Y}(x) = 0$ . This allowed to compare the approaches against a reference, the optimal acceptable operation (obtained from the N-1 SCOPF dispatch).

In order to create the training data  $(X, Y)$ , loads were assumed to be distributed within  $\pm 25\%$  of the nominal loads. The

samples were drawn from a multivariate Gaussian distribution (with a Pearson's correlation coefficient of 0.75 between all load pairs) and converted to a marginal Kumaraswamy(1.6, 2.8) distribution by the inverse transformation method. The generator powers were randomly dispatched in their respective operation limits, such that the total load and total generator power are matching. The final  $(X, Y)$  consisted of 500000 samples with  $p = 65$  features including load levels, pre-fault generation levels and line flows and the binary label, the acceptable/unacceptable operation.

To study the *asym. weighting*, *single- $\varepsilon$*  and *condition-specific- $\varepsilon$*  approaches, a DT was learned via CART [22] by using the package *scikit-learn* 0.18.1 [32] in Python 3.5.2; default settings were used (e.g., minimizing the gini impurity) except the weighting of probabilities of the samples. Whereas in the *single- $\varepsilon$*  and *condition-specific- $\varepsilon$*  approaches, we used balanced weights, we varied the weights in  $[0, 1]$  for the *asym. weighting* approach. Under and over-fitting was handled by grid-searching for the hyper-parameters (i) maximal tree depth  $\{5, 6, \dots, 20\}$  and (ii) maximal number of terminal nodes  $\{20, 40, \dots, 100, 200, \dots, 500\}$  involving 5-fold cross validation and 'f1' score as criterion. The MILP was implemented in Pyomo 5.1.1 [33] and the solver was Gurobi 7.02 [34]. The MILP uses a new operating point (defined by the distribution of loads) and makes decisions for all state variables, such as generator power dispatches, including corrective actions and line flows. Further parameters were  $\beta = 0.001$ ,  $\delta = 0.01$  MW and  $\Delta\varepsilon = 5$  MW. For the *condition-specific- $\varepsilon$*  approach, the optimization was solved in each iteration for 1000 samples ( $X^*$ ).

To study the *risk-averse* approach, we used the AdaBoost algorithm SAMME.R [28] with the default parameters of *scikit-learn* (maximal base estimators  $|\Omega_L| = 10$ , learning rate = 1). Platt Calibration was applied by using 100000 samples ( $X^c, Y^c$ ), 5-fold cross validation and was linearized by using  $|\Omega_W| = 62$  line segments with  $\delta_w = 0.01$  (for  $w > 1$  if  $\mathcal{P}_E^1 \geq 0.5$ ). The trade-off coefficient was varied in  $\alpha = \{0, 0.02, 0.04, \dots, 1\}$ .

### B. Data-Driven Security Rules: Classification Versus Control

We start with showcasing the inappropriateness of the test error rate  $\zeta$  for assessing the suitability of identifying control actions from security rules. As discussed, the control error  $k$  yields a more appropriate metric. To demonstrate the mismatch between the two metrics, an unmodified DT (as illustrated in Fig. 2) with balanced sample weights was used.  $\mathcal{N}^t = 100000$  out-of-sample points were used to compute the test error, which was  $\zeta \leq 0.1\%$ . By applying the unmodified security rules  $\mathcal{N}^* = 100000$  optimized operating points were obtained and the control error  $k = 70\%$  was calculated. Even though the test error  $\zeta \leq 0.1\%$  suggests that the DT is capable of achieving high performance predictions, the DT-based security rules are inappropriate for identifying control actions. This demonstrates, as discussed in Section II, the test error is unsuitable to quantifying security rules for control since the optimization drives operation to unacceptable regions.

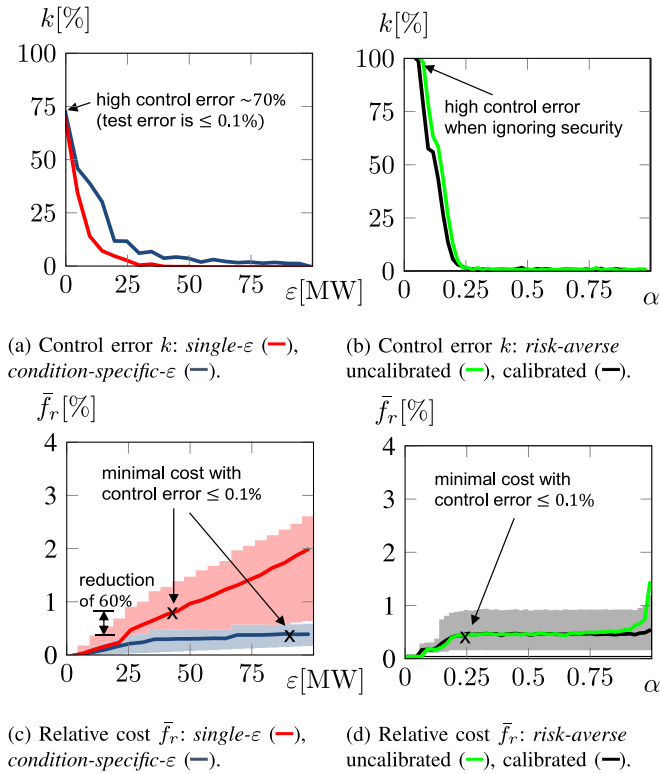


Fig. 5. Cost-risk balance of the discussed approaches: In (a) and (b) is the ‘risk’ represented as control error and in (c) and (d) is the corresponding relative cost based on exclusively comparing the acceptable samples. The shaded regions in (c) and (d) correspond to the 10th and 90th percentiles of the relative cost.

### C. Balancing Cost and Risk

The discussed approaches (illustrated in Fig. 3) were assessed under the lens of the inherent trade-off between risk and cost when using security rules for control. Unless stated otherwise, we computed the control error  $k$  and the average pre-fault operating cost  $\bar{f}_r$  (relative to the optimal reference) with the use of 1000 out-of-sample points ( $X^*$ ). The cost  $\bar{f}_r$  was computed by exclusively comparing the acceptable dispatched samples ( $y^* = 1$ ) against the sample-specific optimal references (SCOPF solution).

By tuning the weights of the samples with respect to the class labels in the *asym. weighting* approach, the lowest control error  $k = 41\%$  has been found at the weight 0.99999 for the acceptable class; the relative cost difference was roughly  $\bar{f}_r = 0.06\%$ . Increasing further the weight for the acceptable class resulted in an empty feasible region. The approach was not capable to obtain a  $k$  close to zero.

The results of the *single- $\varepsilon$*  and *condition-specific- $\varepsilon$*  approaches are presented in Fig. 5(a) and Fig. 5(c). Without any adjustments  $\varepsilon = 0$  MW, a reference solution ( $\bar{f}_r = 0\%$ ) was identified for 30% of the samples (and unacceptable solutions otherwise). By increasing the safety margin  $\varepsilon$ , the control error  $k$  was reduced and the relative cost  $\bar{f}_r$  increased. As discussed, this behaviour is because the estimated decision boundary is shifted to the actual acceptable region (as illustrated in Fig. 3).

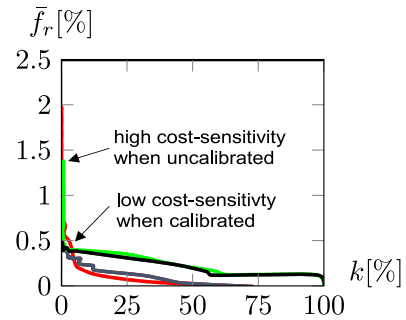


Fig. 6. Sensitivity of the cost-risk balance for the approaches: *single- $\varepsilon$*  (—), *condition-specific- $\varepsilon$*  (—), *risk-averse uncalibrated* (—) and *risk-averse calibrated* (—).

Both approaches were capable of obtaining  $k \leq 0.1\%$ . Since the proposed *condition-specific- $\varepsilon$*  approach tailors separately the safety margin to each condition what results in estimating the acceptable region less conservative, lower operating costs were obtained than in the *single- $\varepsilon$*  approach. In fact, for  $k \leq 0.1\%$ , the result was a reduction in  $\bar{f}_r$  of more than 60% in comparison of using a *single- $\varepsilon$* .

The results of the *risk-averse* approach with and without calibration are presented in Fig. 5(b) and Fig. 5(d). The study included varying the trade-off coefficient  $\alpha$ . By increasing  $\alpha$ , a more risk-averse focus is entailed and the optimized solution is shifted towards regions with higher probability estimates of acceptable operation (to a greener region in Fig. 3(d)). At  $\alpha = 0$  (running a standard DCOPF), the control error  $k$  is 100%. By increasing the coefficient  $\alpha$ , a low control error  $k < 0.2\%$  can be achieved for  $\alpha > 0.25$  and the relative cost  $\bar{f}_r$  increased. For  $0.25 \leq \alpha \leq 0.8$ , the relative cost  $\bar{f}_r$  remains constant for both the uncalibrated and calibrated case. Consequently, for instance  $\alpha = 0.6$  is an appropriate value to balance cost and risk. The main difference between uncalibrated and calibrated case is the cost-sensitivity to high values of  $\alpha$ .

### D. Sensitivity of Cost-Risk Balance

The balance of cost and risk moved along the curves presented in Fig. 6. A typical risk-averse operator aims to achieve low control errors  $k$ . When the parameters  $\varepsilon$  or  $\alpha$  were selected to reduce  $k$ , the cost  $\bar{f}_r$  increased for all approaches. However, this cost sensitivity varies. The *single- $\varepsilon$*  and uncalibrated *risk-averse* approaches particular showed a large increase in cost  $\bar{f}_r$  when  $k \rightarrow 0$ . At each iteration in the *single- $\varepsilon$*  approach, fewer conditions require improvement. Consequently, the cost increases more rapidly than  $k$  reduces. In the procedure to train the ensemble for the *risk-averse* approach, the uncalibrated probability estimates were pushed away from 0 and 1, with many regions having values around 0.5. Although this nonlinear distortion does not impact the accuracy of classifications, it results in wrong probability estimates. The nonlinearity of the distortion is in conflict with the nature of the linear scalarization in the multi-objective optimization (8). In other words, the nonlinear distortion of probability estimates results in more difficulties in tuning  $\alpha$  and leads to higher cost-sensitivities when  $k \rightarrow 0$ . Both



TABLE I  
CONTROL ERROR  $k$  FOR SEEN AND UNSEEN OPERATING CONDITIONS

	<i>asym. weighting</i>	<i>single-<math>\epsilon</math></i>	<i>condition-specific-<math>\epsilon</math></i>	<i>risk-averse (cal.)</i>
seen	41 %	$\leq 0.1$ %	$\leq 0.1$ %	$\leq 0.1$ %
unseen	56 %	$\leq 0.1$ %	45 %	$\leq 0.1$ %

proposed approaches, the calibrated *risk-averse (cal.)* approach and the *condition-specific- $\epsilon$* , showed a reduced increase of  $\bar{f}_r$ , when  $k \rightarrow 0$ . In terms of balancing cost and risk, both proposed approaches outperform approaches of the current literature and resulted in roughly  $\bar{f}_r = 0.5\%$  with  $k \leq 0.1\%$ .

### E. Applicability to Unseen Operating Conditions

As discussed, the approaches deal differently with the trade-off between cost and risk as illustrated in Fig. 3. Even after approach-specific improvements, critical regions might remain unacceptable (marked with X in the figure). Consequently, those approaches would not be applicable to unseen operating conditions. To validate the performance under unseen operating conditions of the approaches, the following study was undertaken: the approach-specific improvements were finalized and  $\alpha$  and  $\epsilon$  with the lowest  $\bar{f}_r$  for  $k \leq 0.1\%$  were selected. The unseen operating conditions were simulated by drawing the generator costs from an uncorrelated uniform distribution in the generator-individual operating limits. Consequently, a very different population of optimized operating points ( $X^*$ ) was obtained and used to calculate the control error  $k$ . 1000 different operating conditions were simulated and the results are shown in Table I. As assumed, the *risk-averse* and *single- $\epsilon$*  approaches were applicable with  $k \leq 0.1\%$  to unseen operating conditions since the improvement does not focus on regions biased in the optimization. As discussed, the *asym. weighting* was not suitable for seen operating conditions, consequently the same counts for unseen operating conditions. The *condition-specific- $\epsilon$*  approach improves iteratively boundaries in regions where the optimized operating points ( $X^*$ ) accumulate; all other decision boundaries are non-improved. Consequently, changing the operating conditions in the optimization drives the operation onto those non-improved decision boundaries resulting in high  $k = 45\%$ .

### F. Computational Feasibility

To finally judge the applicability, we discuss the computational feasibility. In all discussed approaches, 500,000 generated samples were used to train the classifiers; however, it is possible that a much smaller number of simulations is required when combining the proposed work-flow with importance sampling techniques to maximise information gain.

The offline identification of the safety margin  $\epsilon$  (that satisfies a control error  $k \leq 0.1\%$  with lowest cost) required 4000 and 21000 computations of the class labels in the *single- $\epsilon$*  and in the *condition-specific- $\epsilon$*  approach, respectively. In the *single- $\epsilon$*  approach, a simple half-interval search was applied and in the *condition-specific- $\epsilon$*  approach 21 iterations were needed to reach a control error  $k \leq 0.1\%$ . Since the DT had  $|\Omega_T^1| = 82$

TABLE II  
COMPLEXITY OF THE MILP TO BE SOLVED PER CONTROL TIME FRAME

	#constraints	#continuous variables	#binary variables
<i>single-<math>\epsilon</math> / condition-specific-<math>\epsilon</math></i>	775	88	82
<i>risk-averse</i>	147	150	20
<i>risk-averse (cal.)</i>	250	251	21

and  $|\Omega_T^0| = 118$  terminal nodes, the MILP involved 82 binary variables; the full size of the optimization problem is given in Table II. To solve this optimization problem using Gurobi 7.02 [34] needed a pure solver time of less than 0.1 s on a standard laptop for each operating point that was studied.

In the proposed *risk-averse* approach, the trade-off coefficient  $\alpha = 0.62$  that results in a control error  $k \leq 0.1\%$  with lowest cost was identified offline using a half-interval search after 3 steps and involved 3000 computations of the class label. The ensemble had  $|\Omega_L| = 10$  DTs and the MILP involved 21 binary variables in the calibrated case. The pure solver time was less than 0.1 s for each studied operating point.

The problem increases in complexity for larger and more realistic power systems. A large number of samples is required to learn accurate classifiers [3]. However, we estimate the increase in the complexity of the optimization problem will be only moderately higher than the increase in an equivalent OPF problem. Note, all of the aforementioned approaches require a single DT/DT ensemble independent of the number of contingencies considered. Consequently, even if many more contingencies have to be taken into account in a larger system, still only a single DT/DT ensemble is trained and accounted for in the optimization.

### G. Discussion

The key advantage of the proposed approaches over current approaches is the ability to shift computations from the control time frame to the offline time frame (as discussed in the introduction and shown in Figure 1). In both proposed approaches, the *condition-specific- $\epsilon$*  and the *risk-averse* approach, the computation in the control time frame was less than 0.1 s, consisting purely out of the solver time for the single optimization problem. No additional computations are required as all approaches are directly applied to the expected operating point. Both approaches outperformed current data-driven approaches in better balancing cost and risk. Finally, the calibrated *risk-averse* approach performs well for a wide range of values for  $\alpha$  and is robust to unseen operating conditions.

The proposed work-flow generalizes to the operation of power systems where a risk of instability and operation cost must be balanced under operational uncertainty, and is applicable to distribution and transmission grids. The operational uncertainty may include but is not limited to uncertainty in loads and generator outputs, such as wind turbines or photovoltaic panels. Appropriate risk metrics will depend on the application, but they can be flexibly defined through the acceptability criterion as long as it can be described by a binary criterion (e.g.

1 for acceptable operation and 0 for an unacceptable operation of the power system); consequently, the proposed approaches could be used to account for e.g. line overloads, over-voltages or transient stability. Those different risk functions are described by a DT ensemble and could be learned through other supervised machine learning algorithms, such as random forests, extremely randomized trees or other boosting algorithms. Additionally, the cost function that is used in the risk-cost balance could include terms related to the loss of load or undesirable power peaks. Lastly, the approaches presented in this work can be applied to a larger class of operational challenges, including AC (optimal) power flow and unit commitment problems.

The approaches are limited when aiming to obtain a guaranteed security certificate. As discussed, the security boundary is approximated from data and this approximation leads to inaccuracies which leave a certain residual risk. In other words, in our case study, the control error can be guaranteed to attain  $k \leq 0.1\%$ , but cannot be guaranteed to equal zero.

## VI. CONCLUSION

The challenges of embedding data-driven proxies of security within power systems operational models have been presented, showing how such a scheme can suffer from increased control errors in the absence of risk averse measures. In response, we proposed two approaches: introducing contingency-specific safety margins and moving to a risk-averse formulation by leveraging ensemble learning methods. Through case studies on the IEEE 39-bus system, the proposed approaches were shown to achieve superior cost performance while meeting target risk tolerance levels. The risk-averse approach was shown to be particularly robust against a wide range of uncertainties while also imposing very little computational overhead. This work enables, for the first time, the move from traditional classifiers (as proxy descriptors of data-driven security assessment) to more advanced ensemble methods by proposing a novel risk formulation, GDP optimization framework as well as describing the necessary calibration steps. In the future, feature selection will be improved to decrease the offline computational effort.

## REFERENCES

- [1] P. Panciatici, G. Bareux, and L. Wehenkel, "Operating in the fog: Security management under uncertainty," *IEEE Power Energy Mag.*, vol. 10, no. 5, pp. 40–49, Sep./Oct. 2012.
- [2] E. Heylen, W. Labeeuw, G. Deconinck, and D. Van Hertem, "Framework for evaluating and comparing performance of power system reliability criteria," *IEEE Trans. Power Syst.*, vol. 31, no. 6, pp. 5153–5162, Nov. 2016.
- [3] I. Konstantelos *et al.*, "Implementation of a massively parallel dynamic security assessment platform for large-scale grids," *IEEE Trans. Smart Grid*, vol. 8, no. 3, pp. 1417–1426, May 2017.
- [4] L. A. Wehenkel, *Automatic Learning Techniques in Power Systems*. Norwell, MA, USA: Kluwer, 1998.
- [5] V. Krishnan, J. D. McCalley, S. Henry, and S. Issad, "Efficient database generation for decision tree based power system security assessment," *IEEE Trans. Power Syst.*, vol. 26, no. 4, pp. 2319–2327, Nov. 2011.
- [6] E. S. Karapidakis and N. D. Hatziaziyriou, "Online preventive dynamic security of isolated power systems using decision trees," *IEEE Trans. Power Syst.*, vol. 17, no. 2, pp. 297–304, May 2002.
- [7] Y. Xu, Z. Y. Dong, R. Zhang, and K. Po Wong, "A decision tree-based on-line preventive control strategy for power system transient instability prevention," *Int. J. Syst. Sci.*, vol. 45, no. 2, pp. 176–186, 2014.
- [8] I. Genc, R. Diao, V. Vittal, S. Kolluri, and S. Mandal, "Decision tree-based preventive and corrective control applications for dynamic security enhancement in power systems," *IEEE Trans. Power Syst.*, vol. 25, no. 3, pp. 1611–1619, Aug. 2010.
- [9] D. C. L. Costa, M. V. A. Nunes, J. P. A. Vieira, and U. H. Bezerra, "Decision tree-based security dispatch application in integrated electric power and natural-gas networks," *Electric Power Syst. Res.*, vol. 141, pp. 442–449, 2016.
- [10] C. Liu *et al.*, "A systematic approach for dynamic security assessment and the corresponding preventive control scheme based on decision trees," *IEEE Trans. Power Syst.*, vol. 29, no. 2, pp. 717–730, Mar. 2014.
- [11] F. Thams, L. Halilbaic, P. Pinson, S. Chatzivasileiadis, and R. Eriksson, "Data-driven security-constrained opf," in *Proc. 10th Bulk Power Syst. Dyn. Control Symp.*, 2017.
- [12] J. L. Cremer, I. Konstantelos, S. H. Tindemans, and G. Strbac, "Sample-derived disjunctive rules for secure power system operation," in *Proc. Int. Conf. Probabilistic Methods Appl. Power Syst.*, 2018.
- [13] N. Senroy, G. T. Heydt, and V. Vittal, "Decision tree assisted controlled islanding," *IEEE Trans. Power Syst.*, vol. 21, no. 4, pp. 1790–1797, Nov. 2006.
- [14] L. Wang and C. Singh, "Balancing risk and cost in fuzzy economic dispatch including wind power penetration based on particle swarm optimization," *Electric Power Syst. Res.*, vol. 78, no. 8, pp. 1361–1368, 2008.
- [15] F. Xiao and J. D. McCalley, "Risk-based security and economy tradeoff analysis for real-time operation," *IEEE Trans. Power Syst.*, vol. 22, no. 4, pp. 2287–2288, Nov. 2007.
- [16] E. Karangelos and L. Wehenkel, "Probabilistic reliability management approach and criteria for power system real-time operation," in *Proc. Power Syst. Comput. Conf.*, 2016, pp. 1–9.
- [17] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, 1997.
- [18] J. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Adv. Large Margin Classifiers*, vol. 10, no. 3, pp. 61–74, 1999.
- [19] R. Raman and I. E. Grossmann, "Modelling and computational techniques for logic based integer programming," *Comput. Chem. Eng.*, vol. 18, no. 7, pp. 563–578, 1994.
- [20] M. Sun, I. Konstantelos, S. Tindemans, and G. Strbac, "Evaluating composite approaches to modelling high-dimensional stochastic variables in power systems," in *Proc. Power Syst. Comput. Conf.*, 2016, pp. 1–8.
- [21] M. H. Vasconcelos *et al.*, "Online security assessment with load and renewable generation uncertainty: The itesla project approach," in *Proc. Int. Conf. Probabilistic Methods Appl. Power Syst.*, 2016, pp. 1–8.
- [22] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Monterey, CA, USA: Wadsworth Brooks, 1984.
- [23] D. G. Luenberger and Y. Ye, *Linear and Nonlinear Programming*. Berlin, Germany: Springer, 1984, vol. 2.
- [24] E. Balas, "Disjunctive programming and a hierarchy of relaxations for discrete optimization problems," *SIAM J. Algebr. Discrete Methods*, vol. 6, no. 3, pp. 466–486, 1985.
- [25] T. G. Dietterich, "Ensemble methods in machine learning," *Multiple Classifier Syst.*, vol. 1857, pp. 1–15, 2000.
- [26] L. K. Hansen and P. Salamon, "Neural network ensembles," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 10, pp. 993–1001, Oct. 1990.
- [27] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*. Boca Raton, FL, USA: CRC Press, 2012.
- [28] J. Zhu, S. Rosset, H. Zou, and T. Hastie, "Multi-class AdaBoost," *Ann Arbor*, vol. 1001, no. 48109, 2006, Art. no. 1612.
- [29] A. Niculescu-Mizil and R. Caruana, "Obtaining calibrated probabilities from boosting," *Proc. 21st Conf. Uncertainty Artif. Intell.*, 2005, pp. 413–420.
- [30] B. Zadrozny and C. Elkan, "Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers," in *Proc. Int. Conf. Mach. Learn.*, 2001, vol. 1, pp. 609–616.
- [31] A. Pai, *Energy Function Analysis for Power System Stability*. Berlin, Germany: Springer Science & Business Media, 2012.
- [32] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in python," *J. Mach. Learn. Res.*, vol. 12, no. Oct, pp. 2825–2830, 2011.
- [33] W. E. Hart *et al.*, *Pyomo-Optimization Modeling in Python*. Berlin, Germany: Springer Science & Business Media, 2017, vol. 67.
- [34] *Gurobi Optimizer Reference Manual*, Gurobi Optimization, Beaverton, OR, USA, 2016.

**Jochen L. Cremer** (GS'17) received the B.Sc. degree in mechanical engineering in 2014, the B.Sc. degree in electrical engineering in 2016, and the M.Sc. degree in chemical engineering in 2016, all from the RWTH Aachen University, Aachen, Germany. He is currently working toward the Ph.D. degree in the Control and Power Research Group, Imperial College London, London, U.K. His research interests include machine learning and mathematical programming applied to the operation and planning of power systems.

**Ioannis Konstantelos** (M'12) received the M.Eng. degree in electrical and electronic engineering in 2007 and the Ph.D. degree in electrical energy systems in 2013, both from Imperial College London, London, U.K. His research interests include mathematical programming and machine learning techniques applied to the planning and operation of energy systems.

**Simon H. Tindemans** (M'13) received the M.Sc. degree in physics from the University of Amsterdam, Amsterdam, The Netherlands, in 2004, and the Ph.D. degree from Wageningen University, Wageningen, The Netherlands, in 2009. From 2010 to 2017, he was with the Control and Power Research Group, Imperial College London, U.K. He is currently an Assistant Professor with the Department of Electrical Sustainable Energy, Delft University of Technology, Delft, The Netherlands. His research interests include computational methods for power system reliability assessment, statistical learning, and stochastic control for demand response.

**Goran Strbac** (M'95) is a Professor of electrical energy systems with Imperial College London, London, U.K. His current research interests include electricity generation, transmission and distribution operation, planning and pricing, and integration of renewable and distributed generation in electricity systems.