# Evolving Symbolic Model for Dynamic Security Assessment in Power Systems

Francisco S. Fernandes, Ricardo J. Bessa, *Senior Member*, *IEEE*, and João Peças Lopes, *Fellow*, *IEEE*

***Abstract*—In a high-risk sector, such as power system, transparency and interpretability are key principles for effectively deploying artificial intelligence (AI) in control rooms. Therefore, this paper proposes a novel methodology, the evolving symbolic model (ESM), which is dedicated to generating highly interpretable data-driven models for dynamic security assessment (DSA), namely in system security classification (SC) and the definition of preventive control actions. The ESM uses simulated annealing for a data-driven evolution of a symbolic model template, enabling different cooperative learning schemes between humans and AI. The Madeira Island power system is used to validate the application of the ESM for DSA. The results show that the ESM has a classification accuracy comparable to pruned decision trees (DTs) while boasting higher global interpretability. Moreover, the ESM outperforms an operator-defined expert system and an artificial neural network in defining preventive control actions.**

***Index Terms*—Dynamic security assessment (DSA), artificial intelligence (AI), evolving symbolic model (ESM), grid-forming, supervised learning, reinforcement learning.**

## I. Introduction

THE large-scale integration of renewable energy resources (RESs) is leading the power system towards operating conditions characterized by low inertia, high temporal variability of generation, and an increasing number of distributed energy resources (DERs). In control rooms, this change may drastically increase the supervision and control needs of human experts [1], which hinders their task of ensuring a secure and reliable operation.

The online dynamic security assessment (DSA) task perfectly illustrates how the integration of DERs can increase complexity. Security categorization becomes more complex as the system becomes more prone to instability and has more interacting elements of the DER type, whose behavior depends on the primary sources (e.g., wind and solar) and control methodologies (grid-following and grid-forming variants). The number of scenarios that must be evaluated online increases due to the variability and uncertainty of RESs. The increased complexity and the widespread presence of power electronics lead to large and complex state models that turn online DSA into a problem that is too large to be solved by using model-based techniques. Despite the advances in high-performance computing, resorting to time-domain simulation in model-based techniques makes the process computationally very demanding for an online task.

Data-driven techniques have been seen as promising solutions [2], as they can instantly diagnose the system condition for multiple operation scenarios (OSs) with reasonable accuracy. However, this advantage is overshadowed by the black-box nature of many artificial intelligence (AI)-based techniques, which can create algorithmic aversion in human experts. This calls for the development of inherently interpretable AI models, which humans can easily understand and integrate with their domain knowledge. Particularly in the case of isolated power systems, these AI-based techniques have a high potential. This is because human experts frequently look for understandable and secure ways to operate systems that are easy to validate and do not require an extension of the existing domain knowledge.

One of the early attempts of AI in power systems is the expert system (ES) [3]. It is an inference engine (programming logic) that uses a combination of expert-defined rules and factual knowledge to make deductions about the system state and decisions. Despite the extensive application of the ES at the early stages of the DSA [4], it becomes evident that ES can pose challenges due to its inability to ensure near-optimal performance, lack of leverage on available data, and the demanding nature of deriving a large number of rules through arduous reasoning. Knowledge-based systems are combined with data-driven techniques to address the inherent challenges of conventional ES. Initially, such combinations use data-driven techniques, such as decision trees (DTs) [5] and fuzzy inference systems (FISs), to derive the rules of the ES. Later, this experience culminated in the creation of a hybrid ES, which incorporates data-driven techniques or some of its characteristics inside of ES [6].

Despite the aforementioned efforts, techniques mainly led by artificial neural networks (ANNs) and DT grow popular. ANNs are first used in the context of DSA [7]. Since then,

Manuscript received: June 28, 2024; revised: October 25, 2024; accepted: December 5, 2024. Date of CrossCheck: December 5, 2024. Date of online publication: January 1, 2025.

This work was supported by the ENFIELD (European Lighthouse to Manifest Trustworthy and Green AI) project, European Union's Horizon Research and Innovation Programme (No. 101120657). Views and opinions expressed are, however, those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.



F. S. Fernandes, R. J. Bessa (corresponding author), and J. Peças Lopes are with the Institute for Systems and Computer Engineering, Technology and Science (INESC TEC), Porto, Portugal, and F. S. Fernandes and J. Peças Lopes are also with the Faculty of Engineering of the University of Porto (FEUP), Porto, Portugal (e-mail: francisco.s.fernandes@inesctec.pt; ricardo.jbessa@inesctec.pt; jpl@ fe.up.pt).

DOI: 10.35833/MPCE.2024.000478

the classical multi-layer perceptron has evolved into more complex structures, such as convolutional neural networks [8], graph neural networks [9], and other deep learning-based architectures [10]. The classifier/regressor training paradigm has also undergone different changes, from supervised to semi-supervised learning, where the training set is expanded by the addition of other unlabeled samples [11]. The security categorization enabled by ANNs is often integrated into preventive control modules that promote the generator rescheduling/dispatching, either by iteratively comparing the dispatch output with the regressor security quantification [12], or by incorporating ANN-extracted sensitivity factors into the optimization method [13]. Although ANNs exhibit high generalization capabilities, their interpretability poses inherent challenges to human experts during learning and operation phases [14], which may lead to algorithm aversion.

In contrast to an ANN, a DT is inherently interpretable. This is because the space partition performed to achieve a classification or regression task can be easily transformed into "if-then" rules. Among the widely adopted DT in DSA, iterative dichotomiser 3 (ID3) [15] and classification and regression trees (CARTs) [16] stand out. Two other algorithms worth mentioning are the random forest [17], which addresses the overfitting problem of DT, and the optimal classification tree [18], which addresses the problem of obtaining local rather than global optimal DTs by forming the entire DT in a single step. Regardless of the algorithm, the DT node splitting criteria are based on linear inequalities of the dataset attributes (e. g., $X_1<k_1$, $X_1<k_1$ and $X_2<k_2$, $X_1+X_2<k_{12}$, where $X_1$ and $X_2$ are the illustrative dataset attributes; and $k_1$, $k_2$, and $k_{12}$ are the illustrative numerical constants), which means that the attribute search space can only be partitioned linearly. In this sense, the more complex "real-world" attribute relations may be represented by non-linear inequalities (e.g., $X_1/X_2<k_{12}$ or $\sin(X_1/X_2)<k_{12}$), which are not modeled in the DT. Instead, they are merely linearly approximated, which is a limitation. The DT node splitting criteria are used to find preventive control actions, either via their inclusion as constraints in the re-dispatching optimization [19] or by providing indicators of the most effective control variables [15], [16]. Similar to ANNs, the "if-then" rule also serves as a way to iteratively verify the security of a re-scheduled solution [20]. Still, in complex problems, the number of decision rules can grow fairly quickly [21]. This poses a challenge to the human ability to understand the overall model functioning (e.g., key features and governing laws), which is a characteristic known as global interpretability [22]. Thus, strategies, such as tree pruning [21], multi-objective DT model selection [23], and penalty functions related to the DT complexity [24], are commonly adopted to limit the DT depth, which can lead to a complex trade-off between accuracy and interpretability [18]. Moreover, an approach using gene expression programming to classify the transient stability of the system is proposed in [25], which performs better than ANNs.

Other data-driven techniques that are applied to DSA encompass support vector machines, *k*-nearest neighbours, and Bayesian regression [10].

Owing to the enlarged model size and increased complexity, many of the reviewed AI-based techniques impede humans from fully understanding the capabilities and limitations of the model, as well as from developing an accurate mental model of AI. As a result, risk-averse human experts may have restricted confidence in these AI-based techniques. Recent advances in explainable AI (XAI), such as SHapley Additive exPlanations (SHAPs) [26], which aim at providing additional information about the model processes and decisions, are usually denoted as explanations. These explanations generally correspond to features or layer-wise importance factors for a set of specific model outputs. On a different path, to provide a dimension of the interpretability, [27] proposes methods that derive interpretable proxy models, such as trees or rule-based models from black-box models. XAI-based techniques still have different challenges [14], especially in the case of high-stake decision problems, such as the DSA. Specifically: ① the explanations may not always be consistent with the computations of the original model, which may result in the misunderstanding of the operator of the model behavior; and ② XAI is an additional model that needs to be debugged in the development phase and is not error-free. Therefore, as discussed in [14], it is preferable to stick to an inherently interpretable model. Inherently interpretable models, which are characterized by their symbolic and compact nature, enable a straightforward grasp of their internal dynamics. This not only ensures the interpretability but also endows the inherently interpretable models with the capacity to provide articulate explanations.

This study draws the motivation from two previous works [28], [29] of the AI community. Firstly, it stems from using evolutionary search with a list of 65 basic mathematical operations, referred to as the operations vocabulary, to automatically discover machine learning (ML) algorithms from scratch with minimal human intervention, as detailed in [28]. In contrast to [28], this study uses and refines the simulated annealing (SA) algorithm (benefiting from its convergence properties in [29]) while introducing the concept of templates predefined by humans to guide the search process. Secondly, it is inspired by the potential of evolutionary strategies as a method for reinforcement learning (RL) without backpropagation. Specifically, black-box optimization, namely evolutionary strategies, is employed to maximize the rewards, focusing on tuning the weights and parameters of an ANN policy network [30], instead of using a class of RL algorithms based on the Markov decision process formalism and the concept of value functions. This RL-based model is used as a benchmark model in Section IV. The methodology used by the research described in this paper also adopts a meta-heuristic optimization algorithm for adjusting the policy model. However, this work uses an interpretable symbolic model instead of an ANN as a decision agent.

Furthermore, the methodology proposed in this paper shares similarities with the established paradigms in the field of AI, specifically neurosymbolic learning [31] and genetic programming (or symbolic regression) [32]. Nevertheless, neurosymbolic learning often relies on demonstrations from a teaching oracle. Commonly, this teaching oracle is a pre-trained ANN through RL. These AI-based technologies typically use context-free grammars rooted in automata theory

as a foundational policy model. Genetic programming differs from this study by typically employing a less adaptable tree-based representation, introducing higher complexity in human interaction (i.e., more challenging to interpret and modify). Additionally, tree-based genetic programming is less flexible in managing and integrating knowledge that cannot be easily represented within a hierarchical tree format. However, it is important to note that a tree-based representation is not confined to modeling computer programs or symbolic expressions. It can also represent structures, such as circuit topologies in power electronic converters [33]. It is evident that genetic programming has numerous important applications in the domain of power systems. For example, it can be used to evolve interpretable data-driven bids in electricity markets, which is represented as finite automata [34]. In addition, it can be used to estimate distribution system parameters for state estimation when combined with autoencoder-based ANNs [35] and to develop a generic free-form dynamic load model grounded in basic mathematical/physical equations and available measurement data [36].

Finally, this study brings the following contributions to the state-of-the-art DSA.

1) Unlike techniques based on ANN, a novel evolving symbolic model (ESM) is proposed. The ESM has the capability to generate transparent pieces of knowledge, such as statements and rules, thereby ensuring interpretability and facilitating modifications by human experts. In that sense, the ESM enables the creations of different cooperative learning approaches between humans and AI. These creations can be either at the start of the algorithm, where the operator can define an initial template grounded on prior system knowledge or physical reasoning, or during the evolution process to provide additional guidance to the search algorithm. Compared with genetic programming, the ESM allows for more intuitive interaction and broader applicability by accommodating complex relations and knowledge structures, which are difficult to capture in a rigid tree structure.

2) This study illustrates the ability of the symbolic representation to facilitate seamless model adaptation, thereby enabling the performance of diverse tasks. For instance, this study demonstrates the transformation of a security classification (SC) problem by manipulating a symbolic model classifier into a regressor without the need for any re-training. This unique attribute is termed as reflection in [37], denoting the inherent capability of the model for direct and structured knowledge updates, which enhances its accuracy and comprehensiveness. It is worth noting that attaining such adaptability is considerably more challenging when working with ANN [37].

3) Following the framework of [30] and [38], we use RL and combine it with ESM to derive preventive control actions in case of system insecurity. In contrast to [27], the ESM offers the benefit of not necessitating a black-box oracle to be imitated and approximated with a set of interpretable rules.

The remainder of this paper is organized as follows. Section II describes the ESM. Section III outlines the evolution process. Numerical results are presented in Section IV. Conclusions are presented in Section V.

## II. ESM

### A. Knowledge Representation

A symbolic model is an explicit representation of knowledge that combines language items (symbols, e.g., *a*, 1, +, –) and formal languages (words, e. g., variable $var_1$) to construct symbolic primitives and inference rules and create a framework that facilitates the human interpretation of information within the context of the symbolic model. The basic building blocks used in the symbolic model are the symbols, which can be language items of a general-purpose programming language (GPL) or domain-specific language (DSL). This study adopts symbols or keywords (reserved words that have particular meaning to the compiler, e. g., *if*, *else*, *elif*, and *return*) from a GPL. However, it is important to mention that in cases where non-traditional operators are needed to represent knowledge (e.g., in the decision system), a DSL should be created as it enables the use of customized building blocks. A generic example of a symbolic model $\Theta_0$ is shown in (1), which is designed by using the symbols and words from the set $S_0=\{x,y,z,1,2,3,+,\times,=,<,>,and,if,else\}$.

$$\Theta_0:\begin{Bmatrix} if\ x<2\ and\ y>1: z=x\times 3; \\ else: z=x+2 \end{Bmatrix} \tag{1}$$

When the elements of a symbolic model can be mutated through an evolutionary or a stochastic technique, in such way that another symbolic model is created (or evolved) by learning from data (Section III-D), it is referred to as the ESM. The evolution (or mutation) of the primitive symbolic model $\Theta_0$ to model $\Theta_1$ is illustrated by (2). The bold in (2) serves to identify the differences between models of $\Theta_0$ and $\Theta_1$.

$$\begin{cases} \Theta_0:\begin{Bmatrix} if\ x<2\ and\ y>1: z=x\times 3; \\ else: z=x+2 \end{Bmatrix} \\ \Theta_1:\begin{Bmatrix} if\ x<2\ \boldsymbol{or}\ y>1: z=x\div 3; \\ else: z=x+\mathbf{5.1} \end{Bmatrix} \end{cases} \tag{2}$$

The spectrum of symbolic models that an ESM can potentially transform into is constrained by the symbols defined within its domain $S_E$. This domain encompasses all elements found in the primitive model $\Theta_0$ (Section II-B) and those that can be employed in creating new models (Section III). A generic example is given as:

$$\begin{cases} S_E \bigcup \{V_E, N_E, A_E, C_E, L_E, W_E\} \\ V_E=\{a,b,c,d,x,y,z\} \\ N_E=\{\mathbb{R}\} \\ A_E=\{+,-,\times,\div,=\} \\ C_E=\{<,>\} \\ L_E=\{and,or\} \\ W_E=\{if,elif,else\} \end{cases} \tag{3}$$

where $V_E$ is the set of available variables; $N_E$ is the set of available numerical constants; $A_E$ is the set of available algebraic operators; $C_E$ is the set of available conditional operators; $L_E$ is the set of available logical operators; $W_E$ is the set of available system words; and *a*, *b*, *c*, *d*, *x*, *y*, and *z* are the examples of variables.

In theory, an ESM has the potential to embody all possible symbol combinations within its domain $S_E$. However, in practice, the universe of models is constrained by the primitive model, the model template, the evolutionary or stochastic technique that is used to guide the evolution process, and the specific mutations under consideration, as described in Section III.

### B. Primitive Model Design

A fundamental element of the knowledge presentation in ESM is the primitive model, encompassing the definition of the model template (or sketch) and its initialization. The definition of the model template includes the specification of the architecture and all other parts of the model that are not eligible for mutation, which sets clear boundaries for evolution process. Therefore, the unchecked expansion or reduction of the model is prevented, ensuring that it remains capable of performing the designated task. Figure 1 depicts an illustrative example of basic template of ESM, with mutable components in plain lettering and immutable components highlighted in bold.

***if*** {Conditional statements}: {Sequence of action}
***elif*** {Conditional statements}: {Sequence of action}
***else***: {Sequence of action}

Fig. 1. Illustrative example of basic template of ESM.

In this case, the mutable components of the symbolic model, such as conditional statements or sequences of actions, can be fully specified by a human expert, randomly generated by the machine, or collaboratively defined when the operator inputs part of the symbolic model and the other part is randomly generated. Incorporating the human expert into the initialization loop leverages the valuable prior knowledge typically held by experts through their interactions with the environment (e.g., the power system) or their grasp of physical reasoning. This prior knowledge enhances the primitive model, thereby providing a more effective starting point for the learning process and often resulting in improved convergence. Although this may seem like a potential source of bias in the development of the symbolic model, it should not introduce more biases than similar operations carried out in other AI-based techniques, such as the choice of the architecture of the ANN model and the DT depth. This is because the architecture of the ESM develops into the forms that deliver better performance as the learning process progresses.

## III. Evolution Process

This section describes the evolution process of the symbolic model from data, considering both supervised learning and RL settings. In this section, Fig. 2 is used to illustrate the evolution process that creates a DSA classifier, where 0 denotes the secure operation; and 1 denotes the insecure operation. The illustration is drawn from the case study in Section IV. The illustrative example in Fig. 2 considers a primitive model, which is either randomly generated or cooperatively defined by a human, as discussed in Section II-B. This results in distinct initial solutions and evolution patterns. Figure 2 shows different snapshots of the ESM classifier, representing its initial, intermediate, and final states.

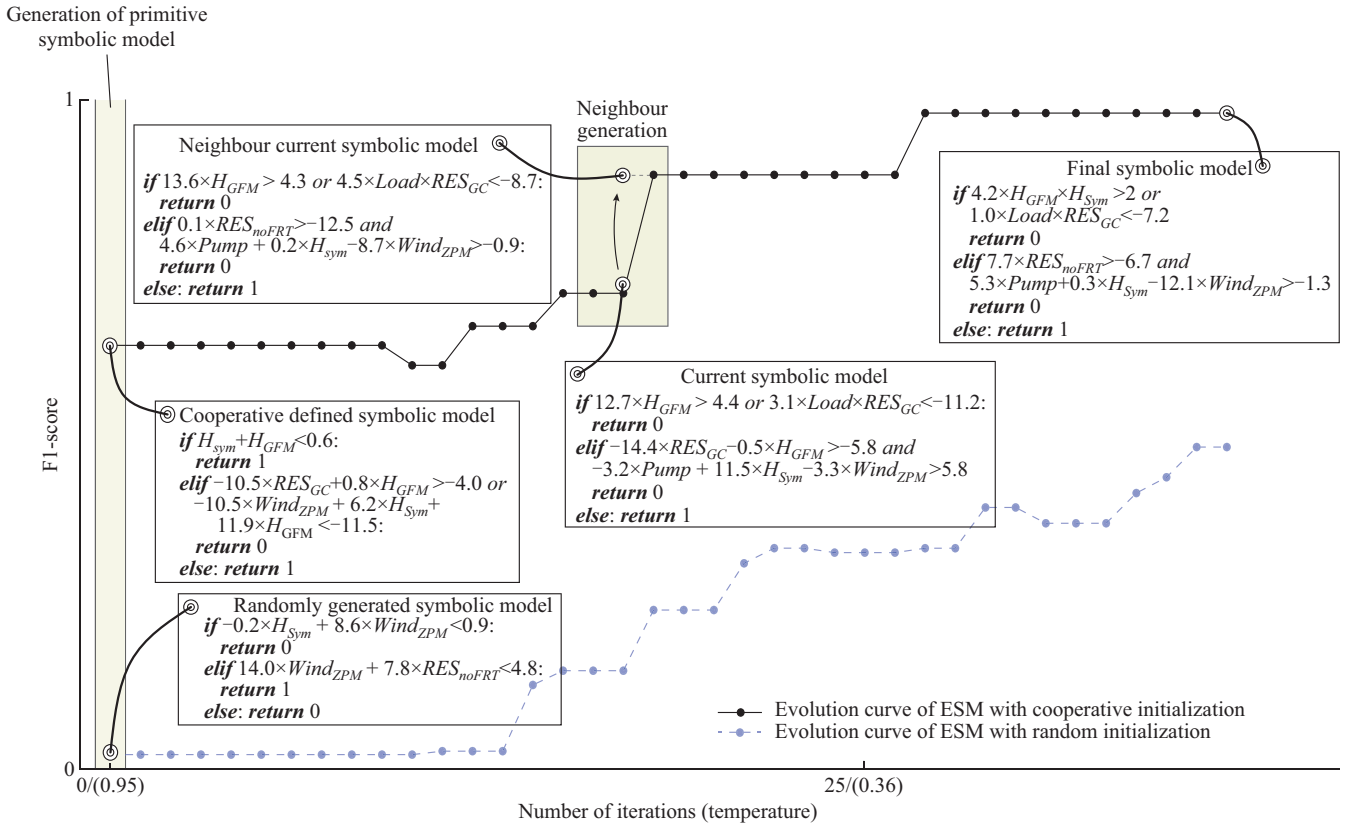


Fig. 2. Illustrative example of evolution process of symbolic model.

### A. Guided Evolution

A parent-child architecture is used to guide the evolution process of the symbolic model, where SA algorithm is the parent that manages the evolution of the symbolic model structure by controlling the mutations, and the child is the differential evolution (DE) algorithm, which is responsible for the tuning of the numerical constants of the program. The SA algorithm enables an effortless evolution control via its temperature equation and neighbourhood definition, removing the need for defining a map between symbols and numeric values (e. g., required in particle swarm optimization), which is a non-trivial task for large symbolic models. The single-solution nature also allows the operator to easily inspect and/or update the ESM at any point during the evolution process without the need to perform any assumption, as with other population methods (e.g., inspect or analyze only the best individual from a population). The DE is a simple, effective, and fast method for tuning the program constants, which displays superiority in accuracy and computational effort compared with other methodologies (e.g., SA algorithm, Bayesian optimization).

The pseudo-code for the evolution process of an ESM is presented in Algorithm 1, where $SM_i$ is the current symbolic model; $SM_n$ is the neighbour symbolic model; $F_i$ is the fitness of $SM_i$ to perform the task of interest; $F_n$ is the fitness of $SM_n$ to perform the task of interest; $T$ is the temperature of the SA cooling scheme; $K_b$ is the virtual Boltzmann constant; $R(0, 1)$ is a uniformly generated random number between 0 and 1; and $T_0$ and $T_f$ are the initial and final temperatures of the SA cooling scheme, respectively.

**Algorithm 1**: evolution process of ESM

**Require**: $K_b$, $T_0$, $T_f$

1. Generate current symbolic model $SM_i$ (Section II-B)
2. Tune numerical constants of $SM_i$ (DE)
3. **while** $T > T_f$ **do**
4. Generate neighbour symbolic model $SM_n$ (Section III-B)
5. Tune numerical constants of $SM_n$ (DE)
6. **if** $F_i > F_n$ or $R(0, 1) < e^{\frac{F_i - F_n}{TK_b}}$ **then**
7. $SM_i = SM_n$
8. Adjust $T$ according to a cooling scheme
9. Perform simplification on $SM_i$ (Section III-C)

The foundational characteristics of the SA algorithm [29] remain visible in the version of symbolic model. During a designated SA cooling scheme (e.g., linear, logarithmic, and exponential), a solution is to pass through different neighbourhood states and change according to the Metropolis acceptance criterion until its energy reaches a minimum value. However, given that the solution is no longer an array of dimension $N$ but a sequence of expressions forming a symbolic model, this version has distinct features, particularly regarding neighbourhood definition (Section III-B). Moreover, it introduces an additional phase, which is known as simplification, during which the symbolic model is inspected and possibly automatically simplified to enhance interpretability (Section III-C).

### B. Mutation-based Neighbour Symbolic Model Generator

In the SA algorithm, a solution may transition into neighbour states, also recognized as candidate states, depending on the fitness of the state and the value of a sampled probability (Metropolis acceptance criterion). A particle neighbourhood can be easily defined in numerical optimization by coupling a distance limit with a random number generator $R(\cdot)$, e. g., $N(X) = X + 0.1R(\cdot)$. Yet, when the solution is no longer an array but a symbolic model, this technique crumbles unless some coordinate scheme is introduced. Therefore, a mutation-based neighbour symbolic model generator is proposed here.

In the mutation-based neighbour symbolic model generator, the neighbour symbolic model is obtained by performing a sequence of mutations sampled from a user-defined mutation pool on a copy of the current symbolic model. To sample the mutations, the value of a uniformly generated random number within the range of 0 to 1 (denoted as $U(0, 1)$) is compared with the probability of adopting each mutation $Pa_j$. If the condition specified in (4) is met, a mutation $m_j(SM_i)$ will be selected. The mutation pool used in mutation-based neighbour symbolic model generator is listed in Table I. The mutations are chosen because they are easy to implement in most of the GPL environments, and they can generate new neighbours, not just simply random solutions.

$$if\ U(0, 1) < Pa_j: SM_n = m_j(SM_i) \quad \forall j \in \mathbf{Z}^+: j \le N_{Mut} \tag{4}$$

where $N_{Mut}$ is the number of mutations.

TABLE I
MUTATION POOL USED IN MUTATION-BASED NEIGHBOUR SYMBOLIC MODEL GENERATOR

| Mutation pool | $Pa_j$ |
|---|---|
| Additional/removal of condition<br>e.g., *if* $x+y>1 \rightarrow$ *if* $x+y>4$ *and* $z<3$ | 1/15 |
| Changes of logical operator<br>e.g., *if* $x+y>4$ *and* $z<3 \rightarrow$ *if* $x+y>4$ *or* $z<3$ | 1/10 |
| Changes of comparison operator<br>e.g., *if* $x+y>4 \rightarrow$ *if* $x+y<4$ | 1/5 |
| Additional/removal of parcels<br>e.g., *if* $x+y>4 \rightarrow$ *if* $x+y-z>4$ | 1/3 |
| Parcel mutation<br>e.g., *if* $x>4 \rightarrow$ *if* $z>4 \mid$ *if* $xz>4 \rightarrow$ *if* $(x/z)>4$ | 1 |

As observed from Table I, the mutations with a larger transformation power (e. g., additional/removal of condition or changes of logical operator) have a smaller probability of adoption, compared with the ones that perform minor changes on the model syntax (e.g., parcel mutation). This enables a frequent generation of neighbours that are sufficiently close to the current solution. Hence, it ensures convergence while allowing the sporadic generation of more diverse neighbours in the name of a larger search space exploration. These probabilities can be determined by trial-and-error experiments in a validation set. The aim is to improve the convergence of the SA algorithm. Alternatively, an optimization approach to the hyper-parameters can be pursued. Figure 2 illustrates the evolution process of a symbolic model when using some of these mutations, where the difference between

the current symbolic model and the neighbour current symbolic model is underlined.

The decision to use diverse and structured mutation pool instead of using a single and more generic mutation (e. g., symbol-for-symbol replacement) is justified by the increase in the controllability of the evolution process of the symbolic model. On the one hand, this eases the implementation process in most of the GPL environments, as it avoids syntax errors that may result from the misplacement of the language operators (e. g., replacing a logical operator with a number). On the other hand, it allows better management of the diversity between the existing and the candidate models, thus ensuring that these are sufficiently close to be considered neighbours, which is essential for method convergence. Regarding the parcel mutation process, it is worth mentioning that particular success is found when using a mutual information (MI)-based variable neighbourhood. Such a neighbourhood is defined at the beginning of the evolution process by computing the MI between all domain variables and the classification labels.

It should be noted that as long as the reasoning for generating sufficiently close neighbours is kept and a similar scheme is adopted when considering the probability of adoption, the user is free to choose different mutations. This means that if the user introduces very transformative mutations from a logical point perspective, lower probabilities should be attached to them, and vice versa. Assuming that the established template and the symbolic domain of the ESM are respected, the human expert can directly perform mutations during learning process, e. g., determining which model parts should mutate. To enhance the operator's understanding of the transformative power of each mutation and ease the generation of close neighbours, the effect of each type of mutation on the output of different expressions can be assessed by means of the Kullback-Leibler (KL) divergence [39]. Since KL divergence measures the difference between two distributions, mutations resulting in a large average KL divergence should be regarded as highly transformative, while those with smaller values should be considered less so.

### C. Simplification Process of Symbolic Model

The symbolic model may evolve into a program featuring redundant statements, resulting in an increase in size and a reduction in overall interpretability. To solve this problem, a periodic model inspection, which is denoted as simplification, is introduced here. The effects that different symbolic model statements have on the symbolic model output are evaluated. Statements that have no effect are removed, and a classical mathematical inspection of each expression is carried out. The considered model template significantly influences this process. Hence, its transference to other problems must be performed with adequate modifications. The simplification process of the symbolic model is described in Algorithm 2.

As illustrated in Fig. 2, the symbolic model maintains complete interpretability throughout the evolution process. This ensures that the operator can readily comprehend the dynamics and decisions of the symbolic model, even during the learning process. Due to the simplification, the constant tuning process becomes faster as simpler models tend to have fewer constants, which is translated into a problem of smaller dimensions for the DE (child) to solve.

**Algorithm 2**: simplification process of symbolic model

1. Output based on simplification process:
2. **for** all statements $St_i$ **do**
3. Evaluate effect of $St_i$ on symbolic model output
4. **if** $St_i$ does not affect symbolic model output **then**
5. Remove part or totality of statements $St_i$
   **end if**
   **end for**
6. Mathematical inspection:
7. **for** all statements $St_i$ **do**
8. Remove model parcels with multi-plications by 0
   e.g., $if(3\times x+0\times y)>1 \rightarrow if(3\times x)>1$
9. Sum all zero-degree polynomials in each expression
   e.g., $if(x+y-3)>1 \rightarrow if(x+y)>4$
   **end for**

### D. Supervised Learning and RL Settings

As illustrated in Fig. 3, the SA algorithm is used to guide the evolution process of ESM of the ESM classifier or regressor in a supervised learning setting by minimizing a loss function and an ESM decision agent in an RL setting via trial-and-error interactions between the simulation environment and the decision agent. The aim is to maximize the reward function.

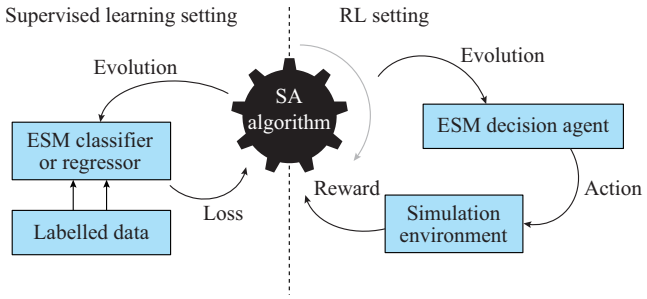


Fig. 3. Evolution process of ESM in supervised learning and RL settings.

In the two learning settings, it may be necessary to adjust the characteristics of the ESM, such as templates and symbolic domains, to align with the distinctive tasks executed by ESM (refer to Section IV-B and IV-D for a more comprehensive discussion). Regarding the SA algorithm, no substantial changes are required, as the only modification pertains to the fitness function defining the ESM solution: either a loss or reward function. The recalibration of the hyper-parameters of the SA algorithm, particularly the virtual Boltzmann constant $K_b$, is required only when large differences in magnitude exist between the loss and reward functions.

Note that the RL adopted in this study does not construct value estimates for particular state-action pairs. Instead, it reasons solely about the value of entire policies [30], [38], by using cumulative reward functions.

## IV. Numerical Results

### A. Case Study

#### 1) Madeira Island Power System

The network is operated at 30 kV/60 kV. In 2024, the generation portfolio consists hydro power (77.17 MVA), thermal power (295.74 MVA), wind power (70.11 MVA), solar power (75 MVA), and waste-to-energy (W2E) power (7.2 MVA). The load range is between 60 MW and 100 MW. To provide flexibility and support to the power system stability with a high share of the RES, the transmission system operator (TSO) relies on reversible hydro pumping stations and a 24 MVA (16 MWh) battery energy storage system (BESS). The share of the RES is expected to grow, leading to challenging OSs due to system stability concerns arising in zero thermal generation scenarios. Thus, the power system is enhanced by installing another 24 MVA BESS and a 15 MVA synchronous condenser.

The Madeira Island power system is particularly significant due to its inherently unstable nature, which is caused by its lack of interconnections. This inherently unstable nature is aggravated by the low inertia and high share of converter-based generation. Consequently, continuous system security monitoring is required, which makes the online DSA fundamental for the control room. Moreover, the presence of diverse DER, such as RES power plants with different capabilities (e. g., fault ride-through (FRT) and participation in frequency control) and BESS that can operate in different control modes, namely grid-forming (GFM) and grid following (GFL), results in a very different response to disturbances. This makes the online DSA a challenging task.

#### 2) Generation of Functional Knowledge

A dataset of 2095 unique OSs is generated by sampling different combinations of load, share of synchronous generation and RES generation. This OS generation is aimed to find scenarios close to the security boundary of the system while preserving sufficient coverage of the system feasible region. For each scenario, the DigsILENT Power Factory software performs merit-order unit commitment, power flow, and dynamic simulation studies in a complete root mean square (RMS) model of the Madeira Island power system. After the stationary studies, an operation point *OP* is established, which is characterized by the system variables listed in Table II. These variables are designed based on prior experience and expert knowledge. Therefore, the recognized correlation with the system stability is established.

The results of the dynamic simulation study, which consists of the simulation of three most severe short circuits to the system stability, are then integrated with the rules of (5). These rules compare the simulation of two frequency stability indices with two protection-related thresholds, enabling a classification for each operation point. The rationale for using frequency metrics as criteria for the SC in the Madeira Island power system lies in the fact that the security loss following severe events is mostly caused by large frequency swings that trigger the load-shedding protections of the power system. The final composition of the generated dataset is 590 (28.2%) insecure operation points and 1505 (71.8%) secure operation points.

TABLE II
System Variables Characterizing Operation Point

| Variable | Parameter |
|---|---|
| $Load$ | System load |
| $Pump$ | Hydro pumping |
| $H_{GFM}$ | System virtual inertia (if BESS is in GFM control) |
| $H_{Sym}$ | System synchronous inertia |
| $RES_{noFRT}$ | RES with no FRT capability |
| $RES_{GC}$ | RES that conforms with system grid code |
| $PV_{ZPM}$ | Photovoltaic (PV) generation with zero power mode (ZPM)-type response during fault |
| $Wind_{ZPM}$ | Wind generation with ZPM-type response during fault |

$$\begin{cases} \Delta f_{CoI} \geq 2 \text{ Hz} \\ \Delta f_{CoI} < 49 \text{ Hz and } Rocof_{CoI}(250 \text{ ms}) < -2.5 \text{ Hz/s} \end{cases} \quad (5)$$

where $\Delta f_{CoI}$ is the maximum frequency deviation of the system center of inertia (CoI); and $Rocof_{CoI}(250\text{ ms})$ is the maximum rate of change of frequency of the system center of inertia, which is measured with a 250 ms sliding window.

### B. Online Dynamic SC

SC, one of the DSA typical functions, is the task of classifying a system operation point as secure (denoted as 0) or insecure (denoted as 1). To address this task in the Madeira Island power system, the aforementioned ESM is used to generate a rule-based classifier. To that end, the primitive symbolic model shown in Fig. 4, which is cooperatively defined with a human expert, is evolved using the instructions of Algorithm 1. In Fig. 4, $K_H$ is the numerical value chosen by the human expert of the system. The primitive version of the classifier includes a well-known stability condition based on the system inertia to illustrate a cooperative initialization in the context of DSA. This knowledge contribution is highlighted in grey in Fig. 4, while the model template is highlighted in bold.

```
#-K_H is chosen by human expert (e.g., 0.5 s) -#
if {H_Sym + H_GFM < K_H}: return 1
elif {Conditional statements}: return K
elif {Conditional statements}: return K
else: return K
```

Fig. 4. Primitive symbolic model of classifier.

Regarding the SA algorithm, $F_i$ corresponds to the classification F1-score in the training dataset and the hyper-parameters, which are chosen through cross-validation. We set $T_i = 0.95$ °C, $T_f = 0.05$ °C, and $K_b = 70$. Regarding the available symbolic domain for the construction/mutation of the classifier, the one defined in (3) is adopted, where the set of variables $V_E$ consists of all the variables described in Table II. It is worth mentioning that despite all these variables during the evolution process of the ESM, not all of them need to be present in the final version of the ESM. In fact, as shown in Fig. 2, the ESM has the ability to retain most relevant variables.

#### 1) Classification Accuracy

To prove the ability of the ESM to create models with

good generalization capacity and to remove its intrinsic randomness, the SA algorithm is run 20 times independently, and the F1-score in the testing dataset is recorded. For each of the 20 runs, different seeds are consistently considered for the SA algorithm and for partitioning of the training/testing datasets. The same study is also conducted using standard and pruned versions of scikit-learn DTs, with minimal cost complexity as the pruned DT model.

All results are compiled into the boxplot of Fig. 5, which shows the distribution of the F1-score of the testing dataset for different SC models. The average training time required for different SC models is also presented in Table III. The depicted times correspond to an implementation in an Apple Mac Mini M2 Pro of CPU with 10 cores and 16 GB of RAM. Note that the average classification time corresponds to the time required to classify 2000 operation points, otherwise its value should be immeasurable.

TABLE III
AVERAGE COMPUTATIONAL TIME REQUIRED FOR DIFFERENT SC MODELS

| Model | Average training time (s) | Average classification time (ms) |
|---|---|---|
| ESM | 5042.00 | 0.55 |
| DT model | 0.01 | 0.67 |
| Pruned DT model | 0.01 | 0.62 |

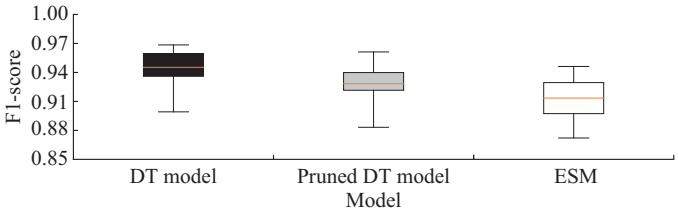


Fig. 5. Distribution of F1-score of testing dataset for different SC models.

While unpruned DT model (which corresponds to DT model) exhibits an accuracy advantage, the results show that ESM can achieve accuracy levels comparable to those of pruned DT model. In fact, ESM can even reach higher accuracy levels for some seeds, while ensuring shorter operation point classification computation time than both benchmarks. Undoubtedly, the ESM requires a longer training time due to the need to perform a step of constant tuning (analogous to a smaller-scale ANN weight adjustment) in every iteration, as the model structure is iteratively changing. Nevertheless, this fact does not constrain the practical adoption of the ESM in SC since the resultant symbolic models are ready for use in an online panorama and can provide very short classification time for thousands of operation points.

*2) Interpretability*

Interpretability in this study refers to understanding the decision-making process of the model, with global interpretability focusing on the overall behavior and local interpretability targeting specific input sets [22]. Global interpretability of a model entails comprehending the principles that govern its behavior across all input ranges, which requires a thorough analysis of the entire model. This is influenced by the complexity of the model, leading to increased mental effort during inspection. Consequently, the total number of mental operations required to inspect the number of total mental operations (TMOs) is a metric for evaluating its global interpretability.

An illustration of the TMO computation for $\Theta_0$ is depicted in Fig. 6. Note that the mental operations required to follow all model decision paths, which are highlighted in grey, include not only arithmetic and logic operators but also every other type of operation that requires mental consideration, such as the selection required when interpreting *if* and *else*.

$$TMO(\Theta_0):\left\{\begin{array}{l} \textit{if}\ (x<2\ \textit{and}\ y>1)\text{: } z=x\times 3\text{;} \\ \textit{else}\text{: } z=x+2 \end{array}\right\}=9$$

Fig. 6. Illustration of TMO computation for $\Theta_0$.

When analyzing the inner dynamics of the model, the human expert does not need to read the entire model. This is because, for some of the inputs, an SC can be achieved through the action of a unique part of the model. The characteristic of the model is the local interpretability, which can still be measured by using the number of mental operations. In this context, the focus is solely on the number of required mental operations (RMOs) to trace the action path of the model for each input. Figure 7 shows the illustration of RMO computation for $\Theta_0(1,0)$ (in view of the input $(1,0)$).

$$RMO(\Theta_0(1,0)):\left\{\begin{array}{l} \textit{if}\ (x<2\ \textit{and}\ y>1)\text{: } z=x\times 3\text{;} \\ \textit{else}\text{: } z=x+2 \end{array}\right\}=7$$

Fig. 7. Illustration of RMO computation for $\Theta_0(1,0)$.

However, using it as a metric to compare the local interpretability of different models can rapidly become complex, as each model can be characterized by multiple values (i.e., one per dataset instance). Thus, the average RMO for all dataset inputs $\overline{RMO}$ is adopted to ease the model comparison process.

The two interpretability metrics, *TMO* and $\overline{RMO}$, are used to score the interpretability of the models generated in the previous subsection (DT model, pruned DT model, and ESM). The results obtained for these metrics are depicted in the radar charts of Fig. 8, where $\overline{RMO}$ is on the right and *TMO* is on the left. Despite their high classification accuracy, the DT model shows very low global interpretability, which is justified by the numerous tree nodes that allow for intense space partitioning and high classification accuracy, as they also represent large sequences of mental operations. However, the pruned DT model can display a much better interpretability without a significant loss of classification accuracy, which puts them very close to the ESM. However, the ESM stands out as the model with the ability to achieve the lowest values in terms of *TMO*, which shows a small edge in terms of global interpretability.

In the case of local interpretability, the differences among three models are small. However, it is clear that the pruned DT model is superior to the other models. If a classification is achieved at the first nodes/conditions of the model, the number of RMOs tends to be higher in the ESM. This is be-

cause conditions in ESMs are usually expressions (e.g., $3\times RES_{noFRT}\times H_{Sym}-RES_{GC}<2$), rather than single variable inequalities (e.g., $H_{Sym}<0.9$), as is the case with the DT model. This architectural dichotomy explains the visible difference between the DT model and the ESM regarding $\overline{RMO}$ values, which is more apparent in the pruned DT model. Still, it is worth mentioning that mathematical expressions of the ESM enable non-linear partitions of the attribute space while preserving a compact nature that often evidences true physical relations.

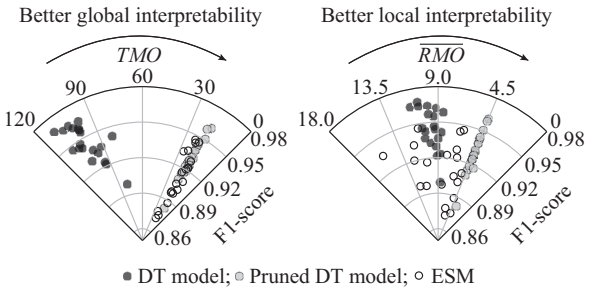


Fig. 8. Interpretability analysis of different models.

To highlight the inherent knowledge value that the generated expressions can carry, two conditions derived from one of the resulting ESMs are briefly analyzed in Fig. 9. To facilitate comprehension, the inequalitys that comprise the conditions are first-hand-manipulated, which makes the important relations more explicit. The range of values that the involved variables can take is also portrayed, with all variables expressed in the per-unit system.

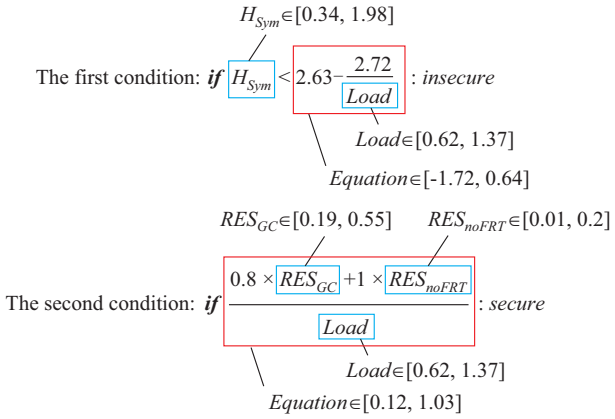


Fig. 9. Analysis of ESM condition interpretability.

As shown in Fig. 9, the first inequality correlates the need for system synchronous inertia $H_{Sym}$ with the system load *Load*. It demonstrates a greater need for system synchronous inertia in higher load scenarios to ensure the security. This correlation makes sense as the load levels that meet the condition often coincide with a high presence of photovoltaic and wind generation. Many generators of these energy sources lack FRT capability or operate in zero power mode (not injecting active power from fault inception until normal operation of the converter). This results in an increased power mismatch after fault clearance. Consequently, it causes a deceleration of synchronous machines after a fault, resulting in frequency nadirs that can trigger system protections (see [40] for more details). By increasing the system synchronous inertia, it is possible to effectively mitigate the rapid frequency variations that occur in these scenarios [40]. Hence, this condition is theoretically sound and practically valid.

The second condition provides an interesting relation among system load, the RES generation that conforms with the grid code, and the RES generation with no FRT capability. Specifically, it establishes a weighted penetration limit for these types of generation based on the load. As previously noted, the presence of a generation with no FRT capability can exacerbate post-fault frequency dips, potentially leading to insecure system conditions [40]. In contrast, when RES generation complies with grid codes, particularly regarding active current injection, the extent of frequency excursions within the islanded grid tends to be mitigated [41], although these excursions remain more pronounced than in grids dominated by synchronous generators. Consequently, it is reasonable that less generation with no FRT capability is permissible compared with the grid code-compliant generation. This is reflected in the larger scalar value associated with $RES_{noFRT}$ in comparison with $RES_{GC}$ in the given condition.

The value of the generated expressions is not only in establishing relations between the right variables in a way that makes sense from a theoretical and practical perspective, but also in the ability to quantify this relation numerically and link it directly to system safety requirements.

### C. Reflection Property: Transformation to Regression

The symbolic and interpretable nature of the ESMs enables seamless direct model updates by the human expert (operator), a characteristic that can be explored not only for initialization/mutation purposes during evolution process (Sections II-B and III-B), but also for augmenting the ESM so that different tasks can be performed.

The model ability, which is known as reflection in [37], is explored to transform the ESM classifier into a regressor. This is because this form provides additional information regarding the distance of a state $OS_i$ to the system security boundary in the form of a security index $S_i$. It should be noted that the method used to create an ESM classifier can also be used to directly generate the regressors. Still, the two-stage approach is undertaken to prove the reflection property that ESMs have.

To make this manipulation (i.e., from classifier to regressor), the output of the security classes (0 denotes secure and 1 denotes insecure) is replaced by the output of a security index $S$, whose computation depends on the existing conditional statements. To achieve the security index expressions, the inequalitys of each conditional statement are manipulated until one of its sides is zero. The non-zero side is then used to determine the security index value when that specific condition is verified. The following example illustrates this manipulation.

Small classifier:
*if* $3\times y<1$: *return* 1
*else*: *return* 0

Small regressor:
*if* $3\times y<1$: *return* $1-3\times y$
*else*: *return* $1-3\times y$

Not that, after this manipulation, the regressor may output the values larger than zero if the system is insecure and output the values lower than zero if the system is secure. Hence, the regressor output of zero should be understood as the security boundary.

The conditional statements defined in this study may include more than one inequality (e. g., *if* ($x<1$ *and* $y<2$)). Hence, in such cases, it is necessary to assess which inequality should be used to define the security index expression. For this purpose, a min/max function is introduced in the regressor output section to select the security index closer to the security boundary (one for each inequality). This transformation is illustrated as follows.

Small classifier:
*if* ($3\times y<9$ and $2\times x>1$): *return* 1
Small regressor:
$S_1=9-3\times y$; $S_2=2\times x-1$
*if* ($3\times y<9$ and $2\times x>1$): *return* $\min(S_1, S_2)$

The above transformations are applied to one of the 20 ESM classifiers generated in Section IV-B. In addition to the modification described earlier, each security index expression is also normalized to the value of the threshold constant of the respective inequality to ensure a similar range of values for the security index, regardless of the conditional statement that verifies. The transformation of ESM classifier into regressor is depicted in Fig. 10.

```
# ---------- ESM classifier ---------- #
if {0.7×H_GFM/Wind_ZPM + 4.03×H_Sym/Load <6.40 and
    3.56×RES_noFRT×H_Sym − 5.87×RES_IP <−2}:
    return 1
elif {−7.87×Load + 3×Load×H_Sym <−8.18}:
    return 1
else:
    return 0
# ---------- ESM regressor ---------- #
# Inequality transformation
S_1 = (6.4 − (0.7×H_GFM/Wind_ZPM + 4.03×H_Sym/Load)) / 6.4
S_2 = (−(3.56×RES_noFRT×H_Sym − 5.87×RES_IP) − 2) / 2
S_3 = (−(−7.87×Load + 3×Load×H_Sym) − 8.18) / 8.18
# Rule-based output
if {0.7×H_GFM/Wind_ZPM + 4.03×H_Sym/Load <6.40 and
    3.56×RES_noFRT×H_Sym − 5.87×RES_IP <−2}:
    return min(S_1, S_2)
elif {−7.87×Load + 3×Load×H_Sym <−8.18}:
    return S_3
else:
    return max(min(S_1, S_2), S_3)
```

Fig. 10. Transformation of ESM classifier into regressor.

Alternatively, a security indicator for the system can be obtained by using dynamic simulation to obtain the security indices described in (5). The distance of these security indices from their limits can then be used as an indicator. In this case, the distance of the frequency nadir from the limit value of 49 Hz ($S_{Nad}=49-f_{Nad}$) is a suitable indicator, as this is proved to be the determining condition among those in (5). Figure 11 displays the normalized value of this nadir-based security indicator $\bar{S}_{Nad}$ and the normalized regressor output $\bar{S}_{Reg}$ for all points in the dataset.

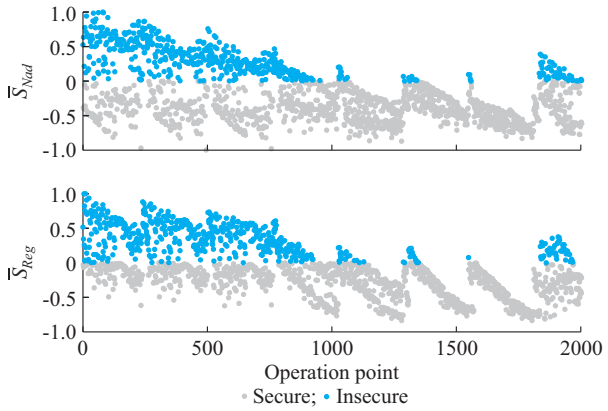


Fig. 11. Scatter plots of $\bar{S}_{Nad}$ and $\bar{S}_{Reg}$.

As shown in Fig. 11, the regressor output is very similar to the nadir-based security indicator, which is computed with a dynamic simulation model of the Madeira Island power system. Hence, it can be concluded that using the manually created regressor to determine the degree of system security is a good approximation of reality.

### *D. Preventive Control*

When the operation point is classified as insecure, the human expert requires a preventive control action that can mitigate system insecurity in the case of any disturbance. In the Madeira Island power system, this preventive control action entails deploying additional synchronous and/or virtual inertia ($H_{sa}$ and $H_{va}$) rather than a more classical approach, such as the generation re-dispatch. This choice is justified by the high share of low synchronous machine scenarios that make: ① small operation point change inefficiently from a security point of view; and ② significant operation point change costly due to the RES curtailment expenses. For this task, Algorithm 1 is used to evolve a rule-based decision agent under an RL setting. This means maximizing a reward function, which is detailed in Section IV-D instead of the F1- score.

The primitive version of the ESM decision agent, which is described in Fig. 12, has minimal operator intervention in its definition, with the main exception being a condition that differentiates the agent actions based on the online virtual inertia. This allows for prioritizing the allocation of extra virtual inertia when capacity is available ($H_{GFM}<1.92$). In Fig. 12, the ESM template is highlighted in bold, and the initial contributions of the operator are highlighted in grey. The value of the additional synchronous inertia $H_{sa}$ or virtual inertia $H_{va}$ suggested by the ESM decision agent is always subjected to a saturation block to ensure that this inertia volume ex-

ists in the system, i. e., $H_{sa}=clip(H_{sa},0,H_{sa_{\max}})$ and $H_{va}=clip(H_{va},0,H_{va_{\max}})$. The available inertia volume ($H_{sa_{\max}}$, $H_{va_{\max}}$) corresponds to the inertia of all synchronous condensers that are offline in the case of synchronous inertia and to the inertia of the GFM control for all offline BESS in the case of the virtual inertia.

***if*** $S$>0:
  ***if*** {$H_{GFM}$<1.92}: $H_{sa}$={Action expression}, $H_{va}$={Action expression}
  ***else***:
    $H_{sa}$={Action expression}, $H_{va}$={Action expression}

Fig. 12. Primitive version of ESM decision agent.

*1) Reward Function*

The synchronous inertia is dispatched via different synchronous condensers (i.e., dedicated or generation units qualified for it), while the virtual inertia is dispatched by changing the control mode from GFL control to GFM control in the BEES. Given that the TSO plans to always have the BESS in operation (for stability support purposes) and no additional expenses result from a change in the battery control mode, the operational cost of a preventive control action is not dependent on the volume of additional virtual inertia. Conversely, the volume of synchronous inertia greatly influences the cost of preventive control actions. In the case of dedicated synchronous condensers, there is a cost associated with the electricity needed to drive the motor. As for the hydro units, there is an additional cost and a preparation time related to the opening of the drainage valves located at the bottom of the turbine chamber to empty it from the water before it can be used as a synchronous condenser.

The ESM for this task is trained under an RL setting. The goal is to maximize the cumulative reward $R$ from (6).

$$R=\sum_{i=0}^{N_{Train}}R_i=\sum_{i=0}^{N_{Train}}[1-0.25(S_i+1)^2-H_{sa_i}K_{to}] \tag{6}$$

where $R_i$ is the environment reward for operation point $i$; $N_{Train}$ is number of operating points in the training dataset; $K_{to}$ is the trade-off between the cost of additional synchronous inertia and the enhanced system security; and $S_i$ is the environment security index for operation point $i$.

The reward function is chosen due to its ability to represent two fundamental features of the problem, namely: ① the non-linear relation between the need to improve further the system security and the current environment security index $S_i$; and ② the costs of dispatching additional synchronous inertia $H_{sa}$. Feature ① is captured by the behavior of the reward function derivative $\partial R/\partial S$, whose form results in the provision of more significant rewards for improvements in unsecured zones ($S_i>>0$), where the action of the ESM decision agent is much needed and poor rewards for improvements in zones are already very secure ($S_i<<0$). Feature ② is captured by using a trade-off function ($H_{sa_i}\times K_{to}$) that reflects how much additional synchronous inertia the operator is willing to dispatch to achieve a sufficient improvement in the system security. The trade-off value ($K_{to}=0.15$) is obtained by inquiring the TSO through an indifference judgment, as explained in [42].

Traditionally, the environment security index $S$ can be attained via time-domain simulation of the reference disturbances. However, this makes the RL procedure extremely heavy in terms of computation. Therefore, to make the procedure feasible, the ESM regressor in Fig. 10 is used as a surrogate model of the system behavior.

*2) Benchmark Models*

1) A simple symbolic model, as depicted in Fig. 13, designed by a human expert with its constants tuned by the DE algorithm are described in Section III-A, which is called Human-ES model. Note that the bold characters in Fig. 13 indicate the model template. The Human-ES model uses different proportional controllers according to the existing amount of virtual inertia $H_{GFM}$. Dividing the model actions based on the online virtual inertia allows for prioritizing the allocation of extra virtual inertia when capacity is available ($H_{GFM}<1.92$).

***if*** ($S$>0):
  ***if*** {$H_{GFM}$<1.92}:
    $\boldsymbol{H_{sa}}=S\times K_1$, $\boldsymbol{H_{va}}=S\times K_2$
  ***else***:
    $\boldsymbol{H_{sa}}=S\times K_3$

(a)

***if*** ($S$>0):
  ***if*** {$H_{GFM}$<1.92}:
    $\boldsymbol{H_{sa}}=S\times 0.91$, $\boldsymbol{H_{va}}=S\times 14.98$
  ***else***:
    $\boldsymbol{H_{sa}}=S\times 11.57$

(b)

Fig. 13. Human-ES model. (a) Initial. (b) Tuned with data.

2) An ANN-based agent constructed using the RL based on evolutionary strategies from [30] is called RL-ANN model. The ANN structure, which is optimized with random search, encompasses four layers with 9, 20, 20, and 2 neurons, respectively, and sigmoid activation functions.

*3) Performance Evaluation*

The RL is executed 20 times, with varying seeds and partitions of the training/testing datasets. The following three metrics are computed for the testing dataset.

1) $R_{Test}$, the reward function value in the testing dataset.

2) $H_{add}$, the volume of additional synchronous inertia dispatched by the ESM decision agent, i. e., $H_{add}=\sum_{j=0}^{N_{Test}}H_{sa_j}$, where $N_{Test}$ is the number of operating points in the testing dataset.

3) $S_{worst}$, the highest value of the security index that can still be found when the ESM decision agent action is considered, i.e., $S_{worst}=\max(S_j)$, $\forall j\in[0, N_{Test}]$

Figure 14 presents the values of these three metrics across each of the 20 runs, computed for the ESM and the two benchmark models. Once more, Fig. 14 shows the ability of ESMs to successfully adapt to different tasks.

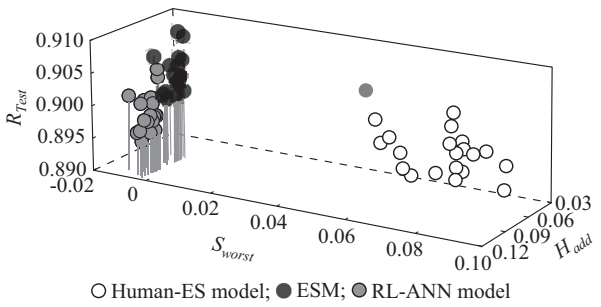


Fig. 14. Performance of ESM and two benchmark models.

As can be seen from Fig. 14, the ESM consistently maintains security across the entire spectrum of the operation points in testing dataset ($S_{worst}<0$), a characteristic not observed in the case of the Human-ES model. This enhancement in security directly influences the reward function, showing that the ESM decision agent does not over-dispatch the volume of synchronous inertia to ensure the system security.

Notably, the ESMs retain full interpretability and compact nature, as seen from Fig. 15, where the final version of ESM decision agent is displayed. Conversely, the RL-ANN model shows very limited interpretability, with an average $\overline{RMO}$ and $TMO$ of 1282 compared with the 12.9 and 18.6 values that characterize the ESM collection, respectively.

This interpretability feature, not displayed by the RL-ANN model, can transform the decision-making process of the operator into a simple sequence of conditions and actions, while still outperforming the RL-ANN model. In the case of the ESM decision agent in Fig. 15, once a state is classified as insecure, then the human expert only has to evaluate the value of the online synchronous inertia $H_{Sym}$ and compare it with the two-term equation that relates the security index $S$ with the online volume of RES generation that conforms with the grid code $RES_{GC}$. After the validation of this condition, the values of additional synchronous and virtual inertia ($H_{sa}$ and $H_{va}$) are directly proposed based on very simple numerical expressions, whose comprehension and even modification due to new experiments or changes in the power system are easily achieved. In real time, this allows for fast and comprehensible computation of the required inertia volume that the system operator easily validates and later dispatches through synchronous condensers and/or batteries in GFM control.

***if*** $S>0$:
  ***if*** $\left\{H_{Sym}<1.13\times S+\dfrac{0.23}{RES_{GC}}\right\}$:
    $\boldsymbol{H_{sa}}=1.32\times\dfrac{PV_{ZPM}}{H_{Sym}}$, $\boldsymbol{H_{va}}=2.02\times\dfrac{Load}{RES_{GC}}$
  ***else***:
    $\boldsymbol{H_{va}}=13.11\times S$

Fig. 15. Final version of ESM decision agent.

As for the time required to train different models, as depicted in Table IV, ESM still stands out as needing a larger time frame; as discussed in Section IV-B, this results from the nuances of its evolution process. Additionally, both symbolic models exhibit shorter decision time when compared with the RL-ANN model, which is a normal consequence of the reduced model size.

TABLE IV
AVERAGE COMPUTATIONAL TIME TAKEN FOR DIFFERENT MODELS IN TRAINING AND DECISION PHASES

| Model | Average training time (s) | Average decision time (ms) |
|---|---|---|
| Human-ES model | 1.99 | 0.68 |
| ESM | 1859.00 | 1.35 |
| RL-ANN model | 1288.00 | 1.79 |

### *E. Discussion: Scalability and Replicability to Other Systems*

The Madeira Island power system displays complex dynamic behavior as a result of the differently controlled elements that coexist in the grid (Section IV-A). The system response to disturbances often exhibits various types of instability, encompassing traditional rotor angle stability issues [43], frequency stability problems that stem from the converter behavior [40], and instability rooted in the converters (itself). These patterns of instability are not exclusive to isolated power systems. In fact, they are becoming increasingly prevalent in interconnected power systems, as shown in [44] and [45]. Therefore, the successful performance of the generated ESMs in the Madeira Island power system instills confidence in the potential applicability of the ESM to DSA in larger systems with slightly different characteristics. This assumption holds if the distinct characteristics of the new systems are captured in the relevant functional knowledge. In other words, the dataset used to generate the ESM in these systems must include instances of the present differentiating stability phenomena, where advanced synthetic data generation methods are essential to extrapolate beyond historical data and create feasible and rare operation point [46].

While the size of the system does not constrain the modeling ability to generate ESMs, it must be highlighted that the final version of ESM may tendentiously become larger for large systems. This is a direct result of complex interactions between more elements and is also expected when using other AI-based techniques (e.g., ANN and DT). Since this influences the global interpretability of the model, the AI-based technique should be applied regarding control areas rather than the whole interconnected system. This enables a suitable application to other systems, allowing a beneficial use of the knowledge interpretation characteristics of the ESMs.

## V. CONCLUSION

This study proposes a novel data-driven model to construct inherently interpretable symbolic models for DSA, using the Madeira Island power system as the proof-of-concept case study.

The ESM, which is developed for the SC task, shows classification accuracy comparable to pruned DT model and better global interpretability. The ESM also thoroughly outperforms the unpruned DT model regarding global interpretability, albeit with a marginal reduction in classification accuracy. The nature of the ESM mathematical expression offers a relevant potential in terms of new knowledge creation and reflection capacity. For instance, the ESM classifier demonstrates versatility by transforming seamlessly into a regressor without requiring retraining. In preventive control actions, it shows high skill in restoring the system security without over-dispatching inertia, outperforming an RL-ANN model and a Human-ES model that responds proportionally to the security index. Importantly, the interpretability and compact nature of the ESM decision agent aligns well with the regulatory requirements for high-risk sectors in the European Union AI Act.

Future work will consist of: ① extending the DSA method to the case of large interconnected power systems; ② ap-

plying the ESM to other tasks in power system control rooms; ③ enhancing the ESM-human interaction by considering, for instance, a higher involvement of the human expert in the hyper-parameters tuning; and ④ combining the SA algorithm with the optimization potential of pre-trained large language models (see [47] for a proof-of-concept).

## REFERENCES


[1] A. Marot, A. Kelly, M. Naglic *et al*., "Perspectives on future power system control centers for energy transition," *Journal of Modern Power Systems and Clean Energy*, vol. 10, no. 2, pp. 328-344, Mar. 2022.

[2] L. Wehenkel, "Machine learning approaches to power-system security assessment," *IEEE Expert*, vol. 12, no. 5, pp. 60-72, Sept. 1997.

[3] Z. Zhang, G. Hope, and O. Malik, "Expert systems in electric power systems – a bibliographical survey," *IEEE Transactions on Power Systems*, vol. 4, no. 4, pp. 1355-1362, Nov. 1989.

[4] R. Christie and S. Talukdar, "Expert systems for on line security assessment – a preliminary design," *IEEE Transactions on Power Systems*, vol. 3, no. 2, pp. 654-659, May 1988.

[5] B. Jeyasurya and S. Venkata, "A knowledge-based approach for power system dynamic security assessment," in *Proceedings of the Third International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, Charleston, USA, Jun. 1990, pp. 645-652.

[6] P. Dash, S. Mishra, M. Salama *et al*., "Classification of power system disturbances using a fuzzy expert system and a Fourier linear combiner," *IEEE Transactions on Power Delivery*, vol. 15, no. 2, pp. 472-477, Apr. 2000.

[7] D. Sobajic and Y. Pao, "Artificial neural-net based dynamic security assessment for electric power systems," *IEEE Transactions on Power Systems*, vol. 4, no. 1, pp. 220-228, Feb. 1989.

[8] A. Gupta, G. Gurrala, and P. Sastry, "An online power system stability monitoring system using convolutional neural networks," *IEEE Transactions on Power Systems*, vol. 34, no. 2, pp. 864-872, Mar. 2019.

[9] J. Huang, L. Guan, Y. Su *et al*., "Recurrent graph convolutional network-based multi-task transient stability assessment framework in power system," *IEEE Access*, vol. 8, pp. 93283-93296, Apr. 2020.

[10] A. Mehrzad, M. Darmiani, Y. Mousavi *et al*., "A review on data-driven security assessment of power systems: trends and applications of artificial intelligence," *IEEE Access*, vol. 11, pp. 78671-78685, Jul. 2023.

[11] R. Liu, G. Verbič, and J. Ma, "A new dynamic security assessment framework based on semi-supervised learning and data editing," *Electric Power Systems Research*, vol. 172, pp. 221-229, Jul. 2019.

[12] R. Sousa, C. Moreira, L. Carvalho *et al*., "Including dynamic security constraints in isolated power systems unit commitment/economic dispatch: a machine learning-based approach," in P*roceedings of 2023 IEEE Belgrade PowerTech*, Belgrade, Serbia, Jun. 2023, pp. 1-6.

[13] J. Fidalgo, J. P. Lopes, and V. Miranda, "Neural networks applied to preventive control measures for the dynamic security of isolated power systems with renewables," *IEEE Transactions on Power Systems*, vol. 11, no. 4, pp. 1811-1816, Nov. 1996.

[14] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature Machine Intelligence*, vol. 1, no. 5, pp. 206-215, May 2019.

[15] L. Wehenkel, T. van Cutsem, and M. Ribbens-Pavella, "Decision trees applied to on-line transient stability assessment of power systems," in *Proceedings of IEEE International Symposium on Circuits and Systems*, Espoo, Finland, Jun. 1988, pp. 1887-1890.

[16] C. Liu, K. Sun, Z. Rather *et al*., "A systematic approach for dynamic security assessment and the corresponding preventive control scheme based on decision trees," *IEEE Transactions on Power Systems*, vol. 29, no. 2, pp. 717-730, Mar. 2014.

[17] M. Lashgari and S. Shahrtash, "Fast online decision tree-based scheme for predicting transient and short-term voltage stability status and determining driving force of instability," *International Journal of Electrical Power & Energy Systems*, vol. 137, p. 107738, May 2022.

[18] J. Cremer, I. Konstantelos, and G. Strbac, "From optimization-based machine learning to interpretable security rules for operation," *IEEE Transactions on Power Systems*, vol. 34, no. 5, pp. 3826-3836, Sept. 2019.

[19] I. Genc, R. Diao, V. Vittal *et al*., "Decision tree-based preventive and corrective control applications for dynamic security enhancement in power systems," *IEEE Transactions on Power Systems*, vol. 25, no. 3, pp. 1611-1619, Aug. 2010.

[20] E. Karapidakis and N. Hatziargyriou, "Online preventive dynamic security of isolated power systems using decision trees," *IEEE Transactions on Power Systems*, vol. 17, no. 2, pp. 297-304, May 2002.

[21] J. Quinlan, "Simplifying decision trees," *International Journal of Man-Machine Studies*, vol. 27, no. 3, pp. 221-234, Sept. 1987.

[22] D. Slack, S. Friedler, and C. Scheidegger, "Assessing the local interpretability of machine learning models," in *Proceedings of 33rd Annual Conference on Neural Information Processing Systems*, Vancouver, Canada, Dec. 2019, pp. 1-7.

[23] A. Bugaje, J. Cremer, M. Sun *et al*., "Selecting decision trees for power system security assessment," *Energy and AI*, vol. 6, p. 100110, Dec. 2021.

[24] D. Lagos and N. Hatziargyriou, "Data-driven frequency dynamic unit commitment for island systems with high RES penetration," *IEEE Transactions on Power Systems*, vol. 36, no. 5, pp. 4699-4711, Sept. 2021.

[25] H. Khattab, A. Abdelaziz, and S. Mekhamer, "Transient security classification using gene expression programming algorithm," in *Proceedings of 15th International Middle East Power System Conference*, Alexandria, Egypt, Dec. 2012, pp. 1-8.

[26] R. Hamilton and P. Papadopoulos, "Using SHAP values and machine learning to understand trends in the transient stability limit," *IEEE Transactions on Power Systems*, vol. 39, no. 1, pp. 1384-1397, Jan. 2024.

[27] C. Ren, Y. Xu, and R. Zhang, "An interpretable deep learning method for power system transient stability assessment via tree regularization," *IEEE Transactions on Power Systems*, vol. 37, no. 5, pp. 3359-3369, Sept. 2022.

[28] E. Real, C. Liang, and D. So, "AutoML-Zero: evolving machine learning algorithms from scratch," in *Proceedings of 37th International Conference on Machine Learning*, Vienna, Austria, Jul. 2020, pp. 1-12.

[29] C. Bélisle, "Convergence theorems for a class of simulated annealing algorithms on $\mathbb{R}^d$," *Journal of Applied Probability*, vol. 29, no. 4, pp. 885-895, Dec. 1992.

[30] T. Salimans, J. Ho, X. Chen *et al*. (2017, Sept.). Evolution strategies as a scalable alternative to reinforcement learning. [Online]. Available: https://arxiv.org/abs/1703.03864

[31] S. Chaudhuri, K. Ellis, O. Polozov *et al*., "Neurosymbolic programming," *Foundations and Trends in Programming Languages*, vol. 7, no. 3, pp. 158-243, May 2021

[32] J. R. Koza, *Genetic Programming: on the Programming of Computers by Means of Natural Selection*. Cambridge: MIT Press, 1992, pp. 450-462.

[33] F. da Silva, R. Glatt, W. Su *et al*., "AutoTG: reinforcement learning-based symbolic optimization for AI-assisted power converter design," *IEEE Journal of Emerging and Selected Topics in Industrial Electronics*, vol. 5, no. 2, pp. 680-689, Apr. 2024.

[34] C. Richter, G. Sheble, and D. Ashlock, "Comprehensive bidding strategies with genetic programming/finite state automata," *IEEE Transactions on Power Systems*, vol. 14, no. 4, pp. 1207-1212, Nov. 1999.

[35] P. Sundaray and Y. Weng, "Alternative auto-encoder for state estimation in distribution systems with unobservability," *IEEE Transactions on Smart Grid*, vol. 14, no. 3, pp. 2262-2274, May 2023.

[36] Y. Lin, J. Wang, and M. Yue, "Free-form dynamic load model synthesis with symbolic regression based on sparse dictionary learning," *IEEE Transactions on Power Systems*, vol. 39, no. 2, pp. 2438-2447, Mar. 2024.

[37] J. Swan, E. Nivel, N. Kant *et al*., *The Road to General Intelligence*. Berlin: Springer Press, 2022, pp. 32-35.

[38] S. Whiteson, *Reinforcement Learning: State-of-the-Art*. Berlin: Springer Press, 2012, pp. 325-355.

[39] J. M. Joyce, *Kullback-Leibler Divergence*. Heidelberg: Springer Berlin Press, 2011, pp. 720-722.

[40] F. Fernandes, J. P. Lopes, and C. Moreira, "Combining batteries and synchronous condensers: the case study of Madeira Island," in *Proceedings of 8th International Hybrid Power Plants & Systems Workshop*, Azores, Portugal, May 2024, pp. 58-64.

[41] P. Beires, C. Moreira, J. P. Lopes *et al*., "Defining connection requirements for autonomous power systems," *IET Renewable Power Generation*, vol. 14, no. 1, pp. 3-12, Jan. 2020.

[42] V. Chankong and Y. Haimes, *Multiobjective Decision Making: Theory and Methodology*. New York: Courier Dover Press, 2008, pp. 25-32.

[43] P. Beires, M. Vasconcelos, C. Moreira *et al*., "Stability of autonomous power systems with reversible hydro power plants: a study case for large scale renewables integration," *Electric Power Systems Research*, vol. 158, pp. 1-14, May 2018.

[44] N. Hatziargyriou, J. Milanovic, C. Rahmann *et al*., "Definition and

classification of power system stability – revisited & extended," *IEEE Transactions on Power Systems*, vol. 36, no. 4, pp. 3271-3281, Jul. 2021.
[45] N. Modi, M. Escudero, K. Aramaki *et al*., "High inverter-based resource integration: the experience of five system operators," *IEEE Power and Energy Magazine*, vol. 22, no. 2, pp. 78-88, Mar. 2024.
[46] A. Bugaje, J. Cremer, and G. Strbac, "Generating quality datasets for real-time security assessment: balancing historically relevant and rare feasible operating conditions," *International Journal of Electrical Power & Energy Systems*, vol. 154, p. 109427, Dec. 2023.
[47] C. Yang, X. Wang, Y. Lu *et al*., "Large language models as optimizers," in *Proceedings of the 12th International Conference on Learning Representations*, Vienna, Austria, May 2024, pp. 1-41.

**Francisco S. Fernandes** received an M.Sc. degree in electrical and computer engineering from the University of Porto, Porto, Portugal, in 2020. He is currently a Ph.D. student in the doctoral program in sustainable energy systems at the University of Porto and a Researcher at the Institute for Systems and Computer Engineering, Technology and Science (INESC TEC), Porto, Portugal, in the Center for Power and Energy Systems. His research interests include dynamic power system simulation, integration of distributed energy resources, and artificial intelligence.

**Ricardo J. Bessa** received the Licenciado degree, the M.Sc. degree, and the Ph.D. degree from the University of Porto, Porto, Portugal, in 2006, 2008, and 2013, respectively. He is the Coordinator of the Center for Power and Energy Systems at the Institute for Systems and Computer Engineering, Technology and Science (INESC TEC), Porto, Portugal. His research interests include renewable energy, energy analytics, smart grid, and electricity market.

**João Peças Lopes** received the Licenciado degree and the Ph. D. degree from the University of Porto, Porto, Portugal, in 1981 and 1988, respectively. He is a Full Professor at the Faculty of Engineering of the University of Porto, Porto, Portugal, and a Researcher and Associate Director at the Institute for Systems and Computer Engineering, Technology and Science (INESC TEC), Porto, Portugal. His research interests include large-scale integration of renewable power sources into power system, power system dynamics and stability, grid expansion planning, supply and system resilience security, and development of dynamic digital twins for stability assessment.