ATTEST

Ref. Ares(2022)7792165 - 11/11/2022

WP4

Predictive Management Tools for the Integrated Operation of Transmission and Distribution Systems

Tool for On-line Dynamic Security Assessment

User Guide

D4.7

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 864298.

DOCUMENT CONTROL PAGE

DOCUMENT	D4.7 – Tool for On-line Dynamic Security Assessment
TYPE	Deliverable
DISTRIBUTION LEVEL	Public
DUE DELIVERY DATE	31 / 08 / 2022
DATE OF DELIVERY	31 / 10 / 2022
VERSION	V1.0
DELIVERABLE RESPONSIBLE	INESC TEC
AUTHOR(S)	Pedro Barbeiro, Carlos Moreira, Fernando Ribeiro
OFFICIAL REVIEWER(S)	Filipe Soares

DOCUMENT HISTORY

VERSION	AUTHORS	DATE	CHANGES
0.1	Pedro Barbeiro, Carlos Moreira, Fernando Ribeiro INESC TEC	10 / 08 / 2022	Template and first version of D4.7 including description of T4.6
1.0	Pedro Barbeiro, Carlos Moreira, Fernando Ribeiro INESC TEC	31 / 10 / 2022	Final version

Table of Contents

List of Figures

List of Tables

Abbreviations and Acronyms

AVR	Automatic Voltage Regulator
ANN	Artificial Neural Network
CoI	Center of Inertia
DSA	Dynamic Security Assessment
FN	Nominal Frequency
FRT	Fault Ride-Through
MLA	Machine Learning Approach
NADIR	Minimum value of frequency reached during the transient period
OP	Operating Point
OPF	Optimal Power Flow
RES	Renewable Energy Sources
RoCoF	Rate of Change of Frequency
SC	Synchronous Condenser
SCOPF	Security-Constrained Optimal Power Flow
SG	Synchronous Generator
TSO	Transmission System Operator
TS	Test Set
WF	Wind Farm

Executive summary

This document presents the deliverable D4.7 - "Tool for On-line Dynamic Security Assessment – User Guide", which is the outcome of task T4.6 of the ATTEST project. The overarching objective of this task was to develop a tool to perform on-line Dynamic Security Assessment (DSA) with respect to frequency stability in future power systems characterized by large shares of converter interfaced generation. To this end, a machine learning approach based on Artificial Neural Networks (ANNs) acts as the "core" of this tool. These ANNs are trained offline using functional knowledge obtained through off-line dynamic simulations for a set of critical contingencies and for all the foreseen operating scenarios. Afterwards, by analysing the frequency indicators (RoCoF and nadir), the tool performs a classification regarding system security (secure/unsecure state) for the operating scenarios under analysis. In addition, it also provides information to the decision-maker about the proper measures to be taken whenever an unsecure state occurs, namely about the synchronous condensers units that need to be turned on to bring the system to a secure state.

The overall methodology of the DSA tool was conceptually defined in deliverable D4.1 [1] of the project, together with its detailed functional specification and description. Differently from D4.1 [1], this deliverable is mainly focused on implementation aspects, aiming at providing users guidance on the fundamental procedures for exploiting the tool. More specifically, the following aspects are addressed in this deliverable: types and formats of input and output data, hardware and software requirements, mathematical model, user guide for the tool execution (which is accompanied by illustrative examples and results). Furthermore, the performance of the tool is also evaluated using a study case. The results attained show that the DSA tool accounts for very promising results regarding estimation accuracy and computational efficiency, thus making it suitable for being used in both offline and online applications.

.

1. Introduction

The main goal of this tool is to perform Dynamic Security Assessment (DSA) with respect to frequency stability in future power systems characterized by large shares of converter interfaced generation. The tool has the inherent capability of assessing system security with respect to network faults that may lead to severe post-fault frequency deviations due to the active power recovery ramps after fault clearance of converter interfaced Renewable Energy Sources (RES). The overall methodology was conceptually defined in the Deliverable D4.1 [1] of the ATTEST project, together with the detailed functional specification and description of the tool.

For this specific purpose, frequency indicators, namely the Rate of Change of Frequency (RoCoF or df/dt) and the minimum value of frequency reached during the transient period (nadir), are taken in consideration in order to assess system security for a given operational scenario and reference contingencies. The rational for the development of the tool relies on inferring the aforementioned frequency indicators using a trained structure based on Machine Learning Approach (MLA). This structure is trained offline exploiting functional knowledge obtained using an off-line dynamic simulation tool of the electric power system for a set of critical contingencies to be identified and covering the foreseen Operating Points (OPs). In case unsecure operating conditions are identified, either by RoCoF and/or frequency nadir violations, it is assumed that the system can be moved to a secure operational domain through the identification of additional synchronous inertia to the system, namely though the selective connection of additional synchronous condensers. Hence, in addition to the classification (secure/unsecure) of the operation scenario, this tool can support the decision-maker by keeping the already dispatched Synchronous Generators (SGs) in operation while bringing on-line Synchronous Condensers (SCs). Figure 1 depicts the high-level functional diagram taken from Deliverable D4.1 [1]. It presents the main functional blocks and a general functional description for each one.

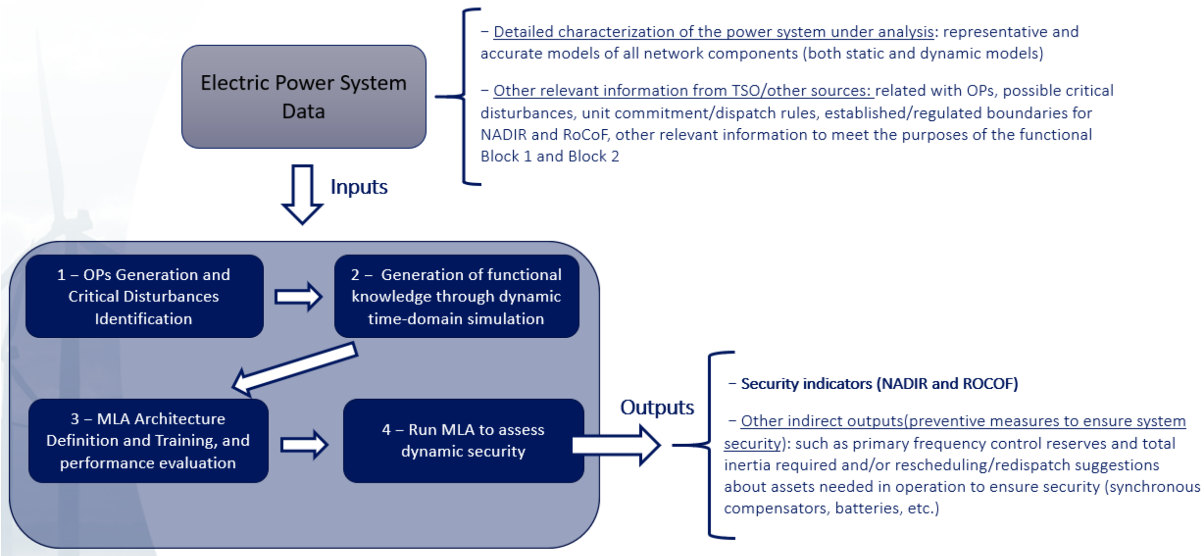


FIGURE 1: FINAL HIGH LEVEL FUNCTIONAL DESCRIPTION OF DSA TOOL (TASK 4.6)

Differently from ATTEST's Deliverable D4.1 [1], this deliverable is mainly focused on implementation related aspects rather than functional or conceptual aspects, aiming at providing users guidance on the fundamental procedures for exploiting the tool. This user guide is therefore accompanied by illustrative examples and results.

2. Study Case

2.1. System Characterization

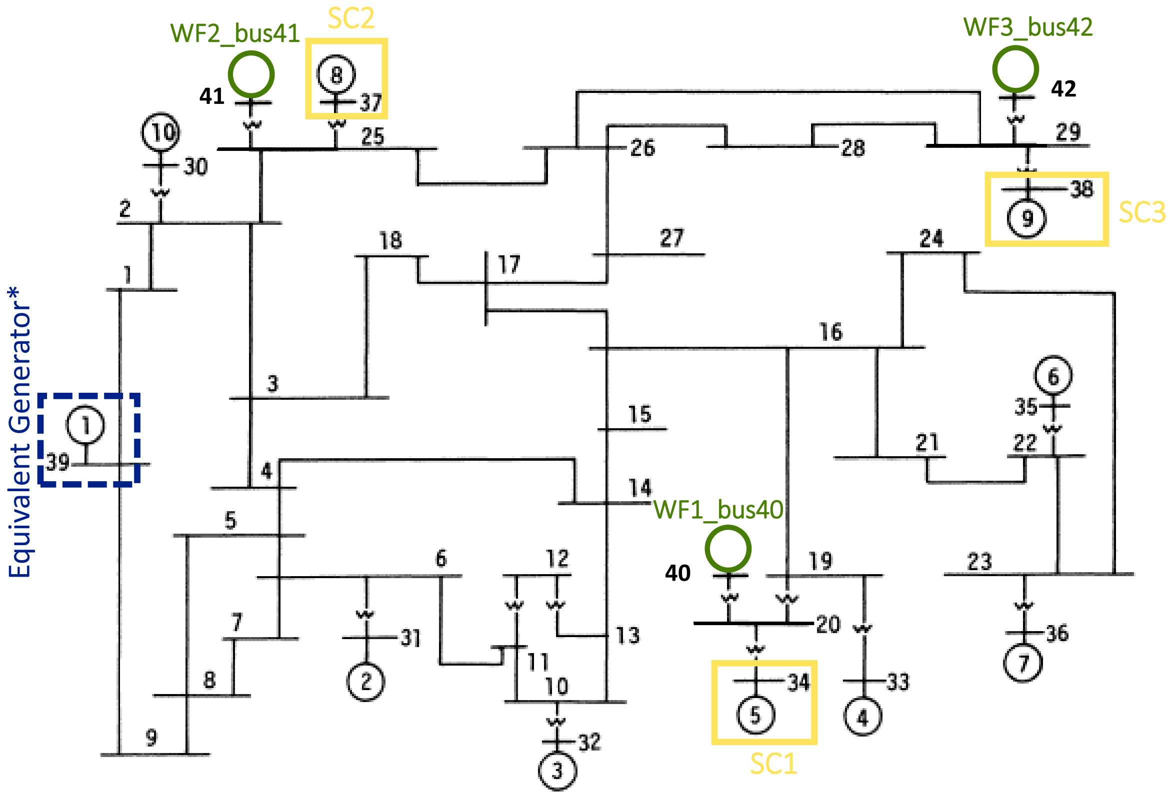
The case study chosen to test the tool was the IEEE 39-bus system, commonly known as "the 10-machine New-England Power System". This system is a simplified model of the high voltage transmission system in the northeast of the U.S.A. (New England area). It was presented for the first time in 1970 and since then it has been widely used by the academic/scientific community as a benchmark system to address problems (testing methodologies and tools) in the transient/frequency stability domain [2]–[3].

The original system consists of 39 buses (nodes), 10 generators, 19 loads, 34 lines and 12 transformers. The nominal frequency of the New England transmission system is 60 Hz and the main voltage level is 345 kV (nominal voltage). The correspondent static and dynamic data can be found in [4] and in [5].

Meanwhile, several modifications have been introduced in studies with different purposes depending on the specificities of the problem that authors are dealing with. In the same way, in order to enable the proof of concept of the DSA tool in the context of the ATTEST project, a few modifications and assumptions related with generation portfolio were also required to be performed, namely an increased RES integration. Regarding the location and size of RES to be integrated in the network, the modifications have closely followed those that were considered in [6]. The main modifications and assumptions introduced in the IEEE 39-bus system in relation to the original data are presented below:

- 3 out of the 10 existing SGs were considered to be decommissioned and refurbished to operate as Synchronous Condensers (SC), which means that they are able to perform reactive power / voltage control and provide inertia to the system (1/3 of the correspondent machine's original inertia was assumed for each SC). Additional information about this technology, namely with respect to the potential benefits that the exploitation of SC might bring to the system in terms of dynamic security can be found in [7]–[10];
- 3 new Wind Farms (WFs) were considered to be installed in locations nearby the decommissioned power plants and having approximately the same rated power: WF1 at bus 40 with 800 MW, WF2 at bus 41 with 700 MW and WF3 at bus 42 with 1000 MW. State-of-art wind generators, more specifically full-size frequency converters (type IV) were assumed to be equipping these WF. This technology implies that the WF are capable of performing reactive power / voltage control within a large range of their P/Q curve and have Fault Ride Through (FRT) ability as well;
- WFs are assumed to be grid code compliant, particularly regarding robustness requirements taking into account FRT and dynamic injection of reactive current during voltage sags. Moreover, the post fault active power ramp recovery was assumed to be 1 MWpu/s. Each WF was connect to the grid through a proper step-up transformer with a rated power sized to the correspondent WF (in MVA) and with a short circuit reactance of 6% (typical value).
- The equivalent generator that represents the remaining interconnected system is modeled by the machine Nr. 1 connected at bus 39. It has a strong impact in the dynamic stability phenomena / frequency excursions due to its size. Thus, aiming at ensuring future scenarios with a very high level of RES integration on generation mix of the remaining interconnect system, it was considered a reduction of about 70% in its total inertia. Note that such scenarios are likely to be the most critical regarding frequency stability problems – the ones that are envisioned to be addressed by this tool.

Based in the above considerations, it is presented in Figure 2 the final single-line diagram of the IEEE 39-bus system, which was adapted from the original presented in [4]–[5] and modified according to the aforementioned assumptions. The main modifications performed over network's infrastructure are highlighted in color and correspond to the 3 new WF (green circles) and to the 3 new SC (yellow squares) that were considered to be integrated in this system.



***Models the remaining interconnected system in terms of all existing synchronous generation units (total inertia) as well as the power flows interchanges with the New England area**

FIGURE 2: SINGLE-LINE DIAGRAM OF THE IEEE 39-BUS SYSTEM MODIFIED TO MEET THE PURPOSES OF TOOL T.4.6 (ADAPTED FROM [4])

2.2. Static and Dynamic Models

This system was modeled in Siemens PTI PSS/E (version 34). The static model used for steady-state power flow calculations is based on a balanced network representation (positive sequence) whose data were taken from [4] and [5]. The new devices included (WFs and SCs) were modeled based on the information stated above. Regarding dynamic models and the corresponding parameters of all the simulated devices, the following were considered:

- Conventional thermal-power plants based on steam turbines were supposed for all SGs, being these units modelled through a round rotor generator model (Quadratic Saturation) – 'GENROU' model [11]. The correspondent synchronous, transient and subtransient parameters required for this model, as well as the inertia time constants were adjusted according to [4].
- The Automatic Voltage Regulators (AVR) are rotating excitation systems of IEEE DC Type 1 according to [4]. This AVR model was thus adopted for all the synchronous generators existing in the system (including SCs), being the correspondent parameters taken from [4].
- Regarding the speed governors, since there is no information available in [4], it was adopted the 'TGOV1' model taken from the PSS/E library [11]. This model, which is a simplified

representation of a steam turbine[1], was adopted for all the thermal SGs considered with exception of SC – this technology cannot perform primary frequency control, therefore do not require governor. The parameters for this model were based on typical values provided in [12].

- For the 3 new WFs admitted in the system, the 'REGCA' model was used for modeling the energy generator, while the 'REECA' was used for modelling the energy electrical model. These models are endowed with FRT capability, being in this way adequate to modelling the full-size frequency converters (type IV) based technology that were assumed to be equipping the existing WFs. In this sense, both models' parameters were tuned in such a way that enables the associated WF to remain connected to the grid during severe voltage sags while injecting reactive power up to its maximum rated power capacity (priority to reactive current injection during voltage sags was given).
- Loads were modelled to have a voltage dependency as follows: a constant current behavior for active power and a constant impedance behavior for reactive power. Frequency dependency and dynamic (time-dependent) load behavior was neglected.

Once more, it is important to note that whenever no information exists in [4] or inconsistent data were detected, typical values available in literature were used to set the parameters of the models mentioned above. In fact, references [12]–[16] were checked carefully and useful information on the referred models, namely regarding typical values for the correspondent parameters were gathered and adopted in a few cases. As the most relevant examples, it can be mentioned the synchronous generation subtransient parameters and speed governors' parameters, where no information was found in reference [4] for these parameters.

[1] The block diagram of this Governor/Turbine system model includes a governor action, reheater time constant, and the ratio of high-pressure turbine.

3. Tool Implementation and Interface

This deliverable addresses the DSA tool *per si* (see block 4 in Figure 1 presented in section 1), being the development of this block the main goal of task T4.6. However, in order to derive the DSA tool, a large volume of off-line simulations over the study case previously described needs to be performed, demanding the use of a power systems dynamic simulation package (which is out of the scope of the developments of task T4.6).

In this sense, the following sections presents a high-level overview of the related methodologies that were envisioned for the blocks 1–3[2] in Figure 1, whereas a more detailed description related with tool modeling and execution is provided for block 4. More specifically, under the scope of block 4, the following aspects are addressed: hardware and software requirements, mathematical model, user guide to the tool execution. Afterwards, in the final section, the performance of the tool is evaluated based on some illustrative results attained for the study case under analysis.

3.1.1. Internal Block 1 – OPs Generation and Critical Disturbances Identification

The main goal of this functional block is the generation of a global dataset with information regarding the behavior of the system in several operating conditions, as well as the identification of the most critical disturbances in the scope of frequency stability. In order to build a global dataset representative of the possible operating conditions expected for the system in an extended time range [3], all the feasible combinations in terms of load, RES and SGs dispatch should considered. The goal is to ensure diversity enough, so the developed MLA based tool can learn the dynamic behavior of the system on an effective and reliable way, i.e., can infer about its dynamic security state based on an accurate estimation of the RoCoF and nadir values.

3.1.1.1. Critical Disturbances Identification

Regarding the identification of the most critical disturbances, for the study case described in section 2, there was no information about incidents that lead the system to critical security regions. In this way, preliminary dynamic simulations were performed to evaluate the frequency response for distinct disturbances and considering extreme operational scenarios.

In the disturbance's selection process, a worst-case approach was followed regarding the type and location of disturbances and the operational scenarios analyzed. In this way, two types of disturbances were considered: three-phase symmetrical short circuits in several different transmission buses (considering different fault clearing times) and power outages – sudden loss of generation (instantaneous tripping of the largest conventional machine). The frequency indicators (RoCoF and nadir) were computed and compared with the correspondent boundaries established/regulated.

For IEEE 39-bus system being studied, due to the absence of data, it was admitted for nadir indicator the boundaries established for the Synchronous Zone of Continental Europe regulated in [17] – 800 mHz (absolute value of the maximum instantaneous frequency deviation to its nominal value Fn[4]). On the other hand, regarding RoCof, since there are not values regulated for this control area, its boundaries were defined taking into account the guidelines of ENTSO-E provided in [18]. According to this reference, the following values are suggested: +/- 2Hz/s for moving average of 500ms window, +/-

[2] The correspondent algorithms were implemented through Python scripts for these functional blocks and can be provided by INESC TEC upon request.
[3] It is assumed that OPs that composed the global database should be capable of representing normal/secure and critical/unsecure operating scenarios.
[4] For the New England case study under analysis the Nominal Frequency (Fn) is 60 Hz.

1,5Hz/s for moving average of 1000ms window and +/- 1,25Hz/s or moving average of 2000ms window. As it can be seen, the RoCoF boundaries vary depending on the moving average window used to calculate them: while more restrictive values are recommended as the time window decreases, there are no consensus about the most proper value to adopted. This problematic is illustrated in Figure 3 where, due to the initial transient variations, it becomes clear that RoCof tends to be higher if computed for shorter average time windows.

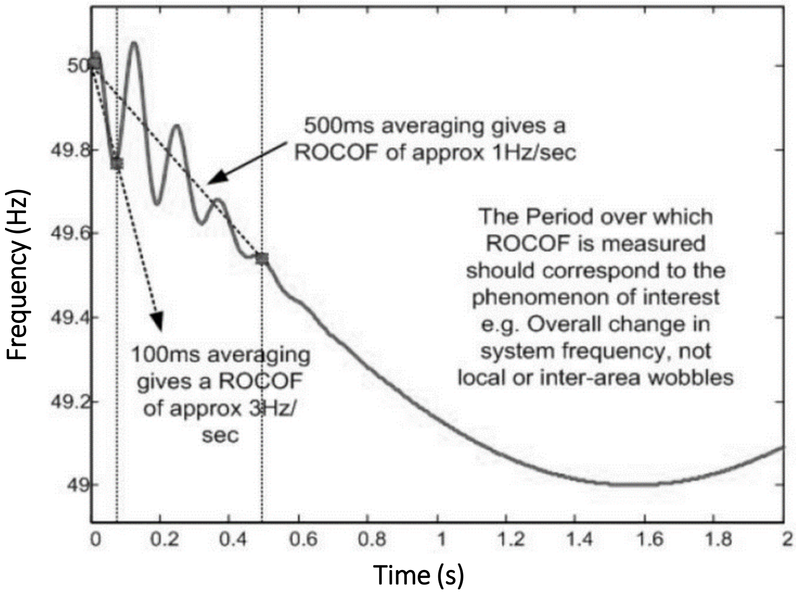


FIGURE 3: ILLUSTRATION OF ROCOF FOLLOWING A LARGE CONTINGENCY EVENT OCCURRING AT 0S IN TIME (MEASURED OVER DIFFERENT AVERAGING WINDOWS) SHOWING THAT A PROPER DEFINITION OF THE ROCOF BOUNDARIES DEPENDS ON THE MOVING AVERAGE TIME WINDOW CONSIDERED [19].

Based on the aforementioned aspects, namely regarding the phenomenon addressed by the DSA tool here proposed (frequency stability), but also the characteristics of the system under study (see section 2 and preliminary results presented below), it was admitted that 1Hz/s would be the most appropriate value to be used as RoCoF boundary.The results of these preliminary studies showed that short-circuits are likely to be the most critical disturbances regarding frequency stability issues (rather than loss of generation for instance). In fact, 3 short circuits locations (when occurring in buses 2, 16 and 31 – see Figure 2) are critical for the system dynamic when considering the referred operating conditions and a fault clearance time of 250 ms[5]. In this sense, these 3 disturbances were the ones selected as the most critical to be simulated in the scope of functional the internal block 2 of this tool (see section 3.1.2).

Figure 4 presents an example of the frequency response attained for a critical operational scenario (with about 3500 MW of load and 70% of wind integration) after a three-phase symmetrical short circuit was simulated in bus 16 with 250 ms of duration (fault begins at 5s in the simulation time).

[5] Typical value when considering failure in the first acting protections and the fault cleared by the backup protections

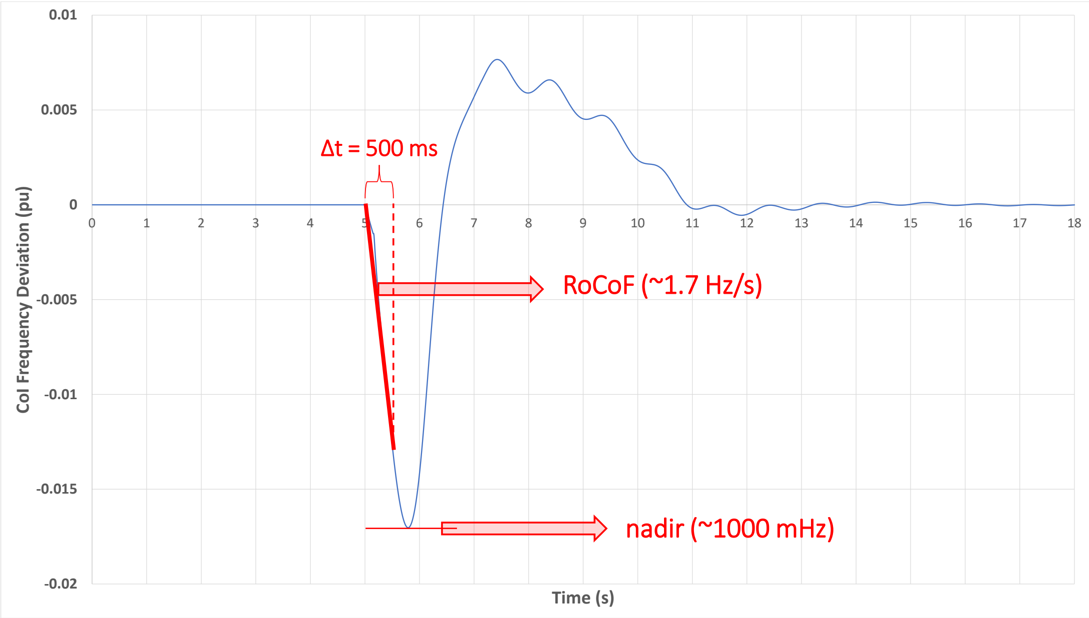


FIGURE 4: ILLUSTRATIVE EXAMPLE OF THE FREQUENCY RESPONSE (DEVIATION IN P.U TO ITS NOMINAL VALUE) FOR A CRITICAL OPERATIONAL SCENARIO AFTER A THREE-PHASE SYMMETRICAL SHORT CIRCUIT OCCURRED IN BUS 16 AT 5S OF SIMULATION AND CLEARED AFTER 250 MS

As it can be seen in Figure 4, over the transient period of the frequency both RoCoF and nadir are violating the regulated boundaries admitted. This result can be easily explained essentially due to the low inertia existing in this operating scenario (high wind power penetration, which together with the response of primary frequency controllers / generators governors adopted dominate the period during fault occurrence. On the other hand, over the post-fault period, are the power recovery ramps defined for wind power converters (set to a 1pu/s rate – see section 2.1) that are likely to be the major factor contributing to this result. For a better comprehension of this phenomenon, it is presented in Figure 7 the active power curves of the 3 WFs dispatched in this scenario where the wind active power recover ramps during pos-fault period are clear evident.

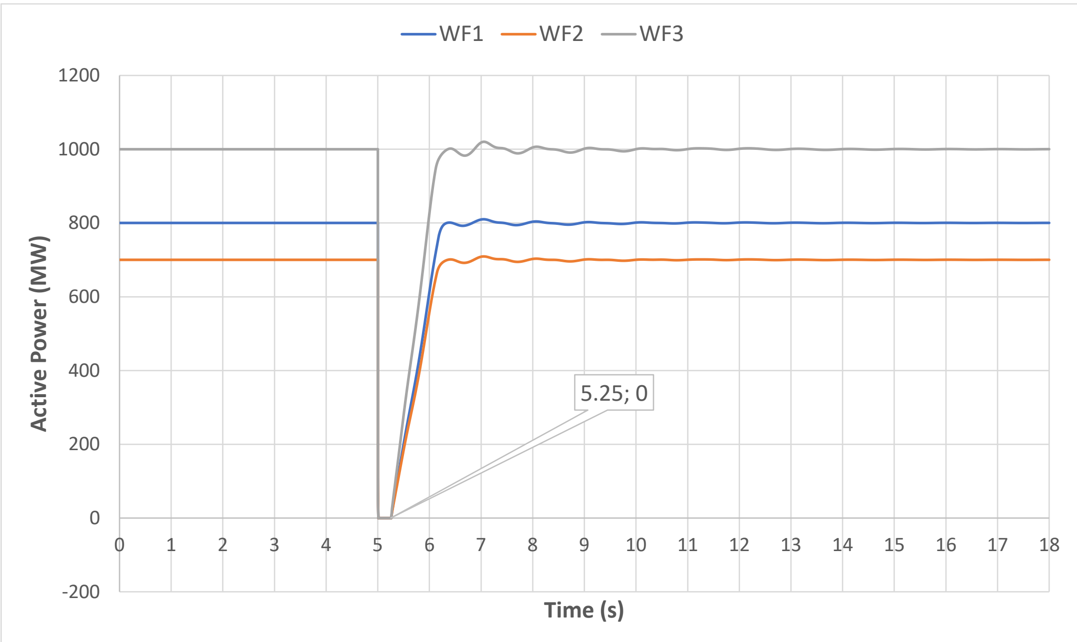


FIGURE 5: ILLUSTRATIVE EXAMPLE OF THE WIND ACTIVE POWER RECOVERY RAMPS POST-FAULT IN A CRITICAL OPERATIONAL SCENARIO. SIMULATED FAULT WAS A THREE-PHASE SYMMETRICAL SHORT CIRCUIT OCCURRED IN BUS 16 AT 5S OF SIMULATION AND CLEARED AFTER 250 MS

Note that, because there is no generation tripping[6], frequency recovers to its nominal value of 60 Hz at steady-state conditions as depicted in Figure 4. Nevertheless, the high values attained for nadir and RoCof would put the system security at risk. In a real case, these values would possibly trigger the under-frequency nadir/RoCof protections (load shedding).

3.1.1.2. OPs Generation Process

Regarding the generation of a global dataset representative of the operating conditions that may be expected in the system, due to the lack of historical information such as historical measurements, market data, or similar, it was followed the procedure described in Deliverable D.4.1 [1] specifically developed to meet this purpose. Having in its basis unit commitment / dispatch algorithms, the developed process generates a large amount of data aiming at ensuring diversity and representativeness for all the possible sets of operating conditions that the system will face. The process is controlled by a set of assumptions and variables related to operational rules (e.g., spinning reserve criteria), dispatch, load and RES integration, and availability of other controllable and noncontrollable devices (e.g., SC) which must be previously defined before its execution. Regarding these aspects, for the specific case of the IEEE 39-bus system under study, the following was considered:

- Total consumption / load levels varying between 2720 MW (valley hours) and 8060 MW for (peak hours). A total of 267 distinct load levels Z were considered, thus resulting in a load step increment ΔX of 20 MW. The power factor of each individual load was kept constant for all the load scenarios generated (in a value equal to the one provided in [4] and [5] for the only load scenario addressed there).
- For each load level, the wind power output was considered varying between 0 MW and the total installed capacity (2500 MW regarding the 3 existing WF), with steps of ΔRES_Wind of 0,4% (corresponds to 10MW in 2500MW), such that all generation wind power buses were increased simultaneously the same percentage regarding the maximum power. This assumption implies that a total of 250 distinct RES levels were considered per each load scenario.
- In relation to the spinning reserve criteria, it was considered a N-1 criterion for the larger synchronous machine dispatched. It is important to note that unit nr. 10 (bus 39), since it represents the equivalent generation present in the remaining transmission connect to the New England area, was not considered in this criterion;
- Regarding the unit commitment algorithm for conventional SGs units, the following merit order was assumed (see Figure 2): unit 10 (bus 39, unit 2 (bus 31), unit 4 (bus 33), unit 7 (bus 36), unit 10 (bus 30), unit 6 (bus 35) and unit 3 (bus 32). The units are scheduled according to this merit until the netload (difference between the total load and the total wind power dispatched) is satisfied and the reserve criteria is met Moreover, it was admitted that at least one conventional SG must be operating in the system. In practical terms, due to the reserve criteria adopted, this assumption implies having a minimum of SGs turned on in the system at any circumstance (i.e., for all OPs generated);

[6] No loss of synchronism was verified for synchronous generators nor wind generation tripped (all WFs considered have FRT capability)

- For the scheduled generating units that resulted of the unit commitment process, the synchronous machines active power dispatch was performed for each OP by distributing the net load proportionally to the generators rated power (in MW) as follows[7]:

$$Pi_{sync_{dispt}} = \frac{Pi_{ratedP\_sync_{dispt}}}{\sum Pi_{ratedP\_sync_{dispt}}} \left(NetLoad + \%P_{Losses_{est}} * PL_{Z_{TOTAL}}\right)$$

where:

- $Pi_{sync_{dispt}}$: Active power dispatch of the *i* SG scheduled [MW].
- $Pi_{ratedP_{sync_{disp}}}$: Rated Power of *i* SG scheduled [MW].
- $NetLoad$: Difference difference between the total load ($PL_{Z_{TOTAL}}$) and the total wind power dispatched [MW]
- $\%P_{Losses_{est}}$: Oversized value for the total active power losses estimated for the system [%]. This value was estimated by carrying out preliminary static analysis (load flow simulations) for the system under study and considering several load conditions, namely high load scenarios. In the specific case of the New England test system under study, this value was set to 3% (in relation to the total load), which corresponds to the largest value attained in the simulations performed.
- $PL_{Z_{TOTAL}}$: Total active load value [MW].

- For each scenario, resulting from the combination of the all the *267* load levels with the 250 wind generation levels (267 x 250 = 66750 operating scenarios), further scenarios were computed by taking into account all the 8 possible combinations regarding the presence or not of the 3 SC considered for the system: without SC, all the 3 SC turned on, only SC1 turned on, only SC2 turned on, only SC3 turned on, SC1 and SC2 turned on and SC3 turned off, SC2 and SC3 turned on and SC1 turned off, SC1 and SC3 turned on and SC2 turned off;

After running the methodology for generating operational scenarios by following the steps as described in Del. D.4.1[1] and taking into account the considerations stated above, a total of 534000 different operational scenarios were generated and the correspondent data compiled in a text file. Basically, this data contains information about the total and individual load values (active and reactive power) and about devices present in the grid (synchronous and wind generation and SC), including its status (in service or out of service) and active power dispatched. In order to build the functional knowledge database required for training the tool, all the OPs generated are evaluated afterwards through dynamic simulations under the internal functional Block 2 (see section 3.1.2).

3.1.2. Internal Block 2 – Generation of functional knowledge through time-domain dynamic simulations

Having the complete static and dynamic model of the power system under study as characterized in section 2, the ultimate goal of this block is to assess all the previously generated OP from a dynamic security point of view through time domain RMS dynamic simulations and taking into consideration the list of contingencies previously identified in the scope of block 1 (see section 3.1.1.1).

At a first stage, each OP resulting from the sampling and dispatching process is accessed in steady state regime to define a credible operating scenario considering grid operational constrains. This process

[7] During the dispatch process of the synchronous generators, it was ensured the minimum technical requirements regarding the active power output of each unit, as well as and the reserve criteria.

supports the definition of the pre-fault operating conditions for each OP that will be used to initialize the dynamic simulation.

Then, for each OP, the predefined list of contingencies is simulated in order to compute the dynamic security indicators (frequency RoCoF and nadir), hence building up and populating a functional database of knowledge that consist of the dispatching information of the grid and the post disturbance frequency indicators. Regarding the computation of the frequency indicators, the concept of Center of Inertia (CoI) frequency as formulated in [20] is considered. ROCOF is calculated considering moving average of 500ms window. As it was explained in section 3.1.1.1, this value is likely to be the most appropriate to assess the phenomenon of interest, i.e., frequency stability related issues due to short-circuits located in critical points of the system and occurring in scenarios with high RES penetration.

It is worth mentioning that the development of the dynamic security assessment tool demands off-line generation of information regarding the power system dynamic performance for a large number of operational conditions and relevant contingencies. Hence, the use of a dynamic simulation package capable of producing the necessary information while taking into account the proper modelling of the generation systems in accordance to the best knowledge of the state is fundamental. On the other hand, the development of dynamic simulation packages for power grid studies is well developed in the market and is out of the scope of the project.

In this sense, for the specific case of the IEEE 39-bus system being studied, all the simulations were performed in PSS/E (version 34)[8]. For all the OP and disturbances in analysis (see section 3.1.1.2), it was considered a total simulation time of 50s and that the fault occurs after 5s of simulation, being cleared after 250ms. Due to the heavy computational burden associated to dynamic simulations and considering the huge number of cases to be simulated (534000 cases per disturbance considered), the whole process was automated through a Python programmed script. By making use of the PSS/E Python automation features, the developed script largely contributed to the computational efficiency of the overall process.

After running all the simulations for all the OPs and the 3 identified contingencies, the most relevant results from dynamic simulations, as well as data related with the operating conditions simulated were gathered and compiled for each pair of OP and simulated contingency and saved in a database. The data includes primary variables and stability indicators which were carefully selected considering the knowledge of the problem under analysis. In particular, the recommendations suggested by the authors in [21] and [22] that address similar problems were taken into account. Afterwards, the following type of information was included in the generated database:

- Characteristics related to the OP conditions (primary variables): active produced powers (per generation unit and SC); aggregated active consumed powers in load buses; aggregated values regarding total active powers categorized by the technology: synchronous, non-synchronous (renewable or non-renewable based power electronics converters); spinning reserve available (synchronous and non-synchronous); total SGs inertia; SC inertia per machine.
- Stability indicators: frequency stability indicators (nadir and RoCoF).

The dataset composed by the variables listed before (primary characteristic variables and grid frequency indicators) can be seen as functional database knowledge that describes the dynamic

[8] Any other Power System Analysis/Simulation Software, being professional, noncommercial (open source) or user based developed may be used as long as they fulfill the requirements of DSA tool proposed regarding the dynamic simulation aspects.

behavior of the system under study. As it will be explained in the following section, in the scope functional block 3, this dataset, after being processed by applying feature selection / extraction techniques will allow defining a proper structure for the MLA chosen, being also used for training and testing purposes.

Figure 6 depicts an illustrative example that intends to show the magnitude of the results attained for the frequency stability indicators (nadir and RoCoF). It is a scatter chart where nadir was plotted as function of RoCoF for all the 534000 cases analyzed and considering a short-circuit simulated at bus 16 with 250ms of time duration. The frequency stability boundaries adopted (see section 3.1.1.1) were also highlighted in red, allowing to compute the correspondent secure and unsecure regions of operation as shown respectively in green and red.

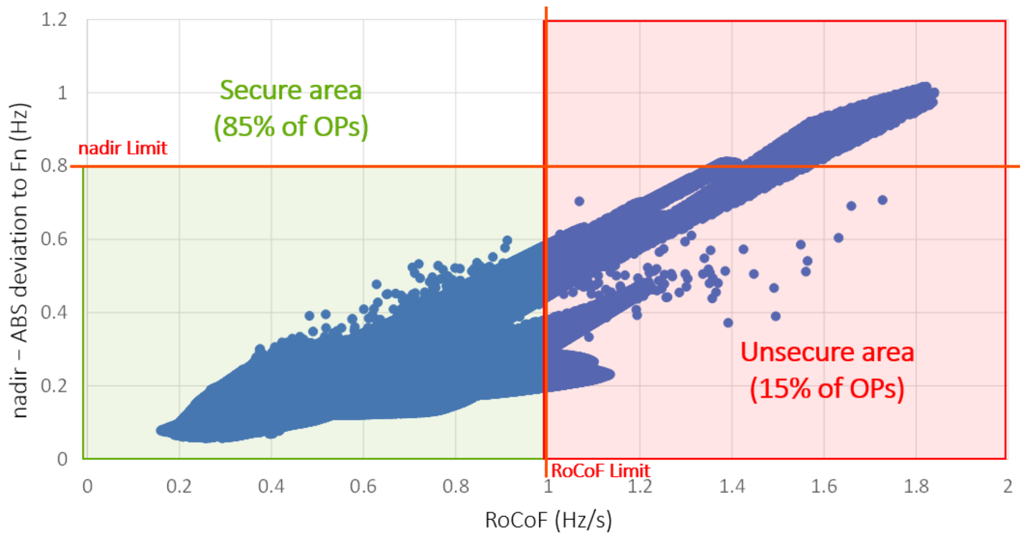


FIGURE 6: EXAMPLE OF THE NADIR AND ROCOF RESULTS ATTAINED FOR ALL THE OPS SIMULATED AND CONSIDERING A THREE-PHASE SYMMETRICAL SHORT CIRCUIT OCCURRED IN 16 WITH A CLEARANCE TIME OF 250 MS

As it can be seen in Figure 6, nearly 85% of the OPs analyzed fall in a secure region, whereas 15% are prone to bring the system to a dynamically unsecure region regarding frequency indicators. As expected, nadir and RoCoF have a strong correlation, being it almost linear. Moreover, it was verified that the large majority of the OPs in the unsecure area correspond to low load scenarios with high wind integration. This result was expected due to the lower synchronous inertia present in the system in these scenarios. Finally, it is important to state, after dispatching the SC available in the system, it was possible to bring all the unsecure OPs to a secure region.

3.1.3.Internal Block 3 – Characterization of the MLA and of the Training Process

As early mentioned at the introduction section (see Figure 1), the DSA tool developed relies on the exploitation of a MLA to assess system security in terms of frequency stability based on the estimation of the frequency indicators RoCoF and nadir (prior defined as output variables of the MLA). To this end, the MLA learns offline the dynamic behavior of the system through the information provided in functional knowledge database generated in Block 2 (see section 3.1.2). This information respects to a set of critical contingencies identified and that were simulated for all the foreseen OPs.

In this sense, the first step is to identify a suitable MLA meeting the aforementioned purposes. There are several MLA such as Support Vector Machine, Artificial Neural Networks, Decision Tree (DT) and Kernel Regression Tree (KRT), among others that can be used in the context of this tool, each one with several advantages and disadvantages. In the selection of the MLA to be adopted, some preliminary analysis over the literature were carried out. As result of this research, the MLA chosen is an Artificial Neural Network (ANN) due to the very promising results in the context of the frequency stability problems as demonstrated in several publications [22]–[24].The next step is to define a proper architecture for the ANN model. Firstly, it is necessary to identifying the most appropriate explanatory features (input variables) to be integrated in the model based on the candidate variables existing in the functional database. The detailed description of this step is presented in section 3.1.3.1.

Afterwards, it is required the definition of aspects related with the ANN internal structure/architecture, namely the number of hidden, layers, neurons, type of activation function, standardization process, etc. Finally, it is mandatory the selection of a proper training method, as well as tunning correctly its parameters. The implementation details of these and other related aspects are provided in section 3.1.3.2.

3.1.3.1. Feature Selection Method

An accurate and time efficient ANN based model, relies at a first stage on the selection of the “best” input variables among all the possible candidates that resulted from the functional knowledge database generated in the internal Block 2 (see section 3.1.2).

For this purpose, a pre-processing method based on so-called feature selection / extraction techniques, was developed. With the aim of removing redundancy and at the same time reducing the problem dimensionality, the proposed method integrates on its basis a F measure of separability technique [25] together with statistical correlation functions. Similar methods have been successfully applied by some authors, such as in [26] and [27], to build hierarchical classifiers for assessing transient stability based on the combination of different Pattern Recognition structures, being the F measure technique usually integrated at the first hierarchical level in these classifiers.

After being adapted for the purpose of this tool, the F-measure based algorithm developed to perform feature selection is as presented in Figure 7. In Table 1 are characterized, in terms of short name, meaning and units adopted, all the candidate variables that were evaluated within this algorithm and that composed the *feature_list* array (see Figure 7).

TABLE 1: CHARACTERIZATION OF ALL CANDIDATE VARIABLES EVALUATED WITHIN THE F-MEASURE BASED ALGORITHM DEVELOPED (VARIABLES THAT RESULTED FROM THE FUNCTIONAL KNOWLEDGE DATABASE GENERATED IN THE SCOPE OF THE INTERNAL BLOCK 2).

Candidate Variables / Features (short name)	Full Name / Description	Units
Load_Total_P	Total Load Consumption (active power)	MW
SGs_Total_Pgen	Total SMs Active Power Production	MW
SG1_Bus39_Pgen	Active Power Production of SG1	MW
SG2_Bus31_Pgen	Active Power Production of SG2	MW
SG3_Bus32_Pgen	Active Power Production of SG3	MW
SG4_Bus33_Pgen	Active Power Production of SG4	MW
SG6_Bus35_Pgen	Active Power Production of SG6	MW
SG7_Bus36_Pgen	Active Power Production of SG7	MW
SG10_Bus30_Pgen	Active Power Production of SG10	MW
WFs_Total_Pgen	Total WFs Active Power Production	MW
Total_SR	Total Spinning Reserve Availabe in the System	MW
Total_H_SMs	Total SGs Inertia	s
H_SC1_Bus34	SC1 Inertia	s
H_SC2_Bus37	SC2 Inertia	s
H_SC3_Bus38	SC3 Inertia	s

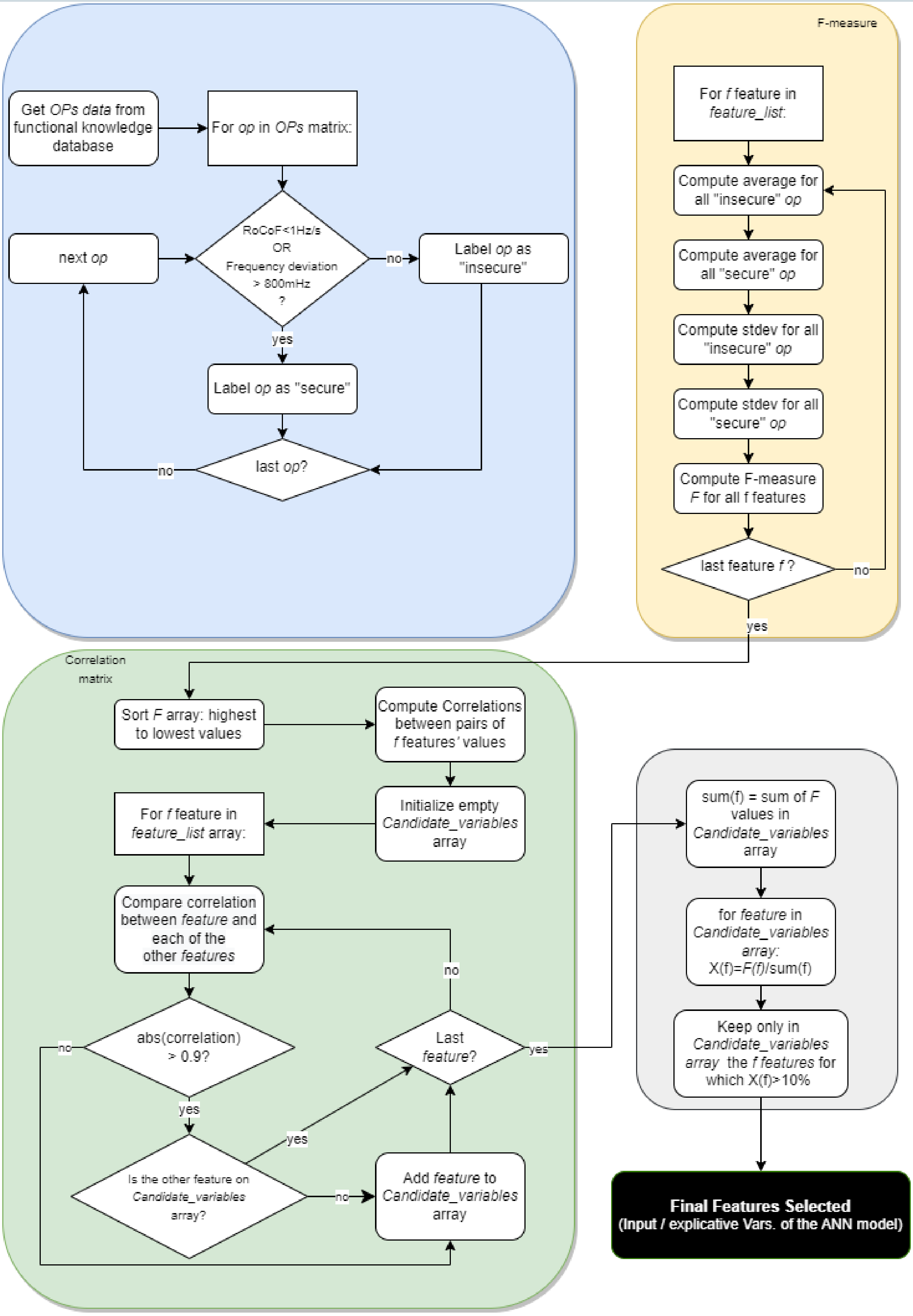


FIGURE 7: FLOWCHART OF THE FEATURE SELECTION ALGORITHM DEVELOPED

The algorithm can be described in five major steps:

- **OPs security classification** (blue): each OP of the functional knowledge database is classified as "secure" or "unsecure" regarding the RoCoF and nadir values;
- **F-measure or *F*-score $\boldsymbol{F_f}$** (yellow) for each feature *f* in *feature_list* array, F_f is computed as follows:

$$F_f = \frac{\left|\bar{x}_{f,secure} - \bar{x}_{f,insecure}\right|}{\left(\sigma_{f,secure} + \sigma_{f,insecure}\right)}$$

Where $\bar{x}_{f,secure}$ and $\bar{x}_{f,insecure}$ are respectively the average of secure and unsecure values of the feature *f*, and $\sigma_{f,secure}$ and $\sigma_{f,insecure}$ are respectively the standard deviation of secure and unsecure values of the feature *f*.

The intuition of using the F-measure (*F* score) is that features with higher values usually have a good discriminative power, i.e., features that have a very distinct average value for "secure" and "unsecure" OPs. In this sense, by ranking the features based on this value, it is possible to have a first guess of the most appropriate ones to be selected.

- **Correlation matrix** (green) between each pair of features *f1* and *f2*, is computed as follows:

$$Correlation(f1, f2) = \frac{\sum\left(x_{f1} - \bar{x}_{f1}\right)\left(x_{f2} - \bar{x}_{f2}\right)}{\sqrt{\sum\left(x_{f1} - \bar{x}_{f1}\right)^2 \sum\left(x_{f2} - \bar{x}_{f2}\right)^2}}$$

Where x_{f1} and x_{f2} are respectively the values of the feature *f1* and *f2*, $\bar{x}_{f1}$ and $\bar{x}_{f2}$ are the average values of the features *f1* and *f2*.

Note that the flowchart the *F* array is sorted from highest to lowest F-measure score, and in the first iteration of the block *"compare correlation between feature and each of the other features"*, the feature with the highest F value necessarily ends up being included in the *Candidate_variables* array.

This step of the algorithm is intended to keep the model as simple as possible. That is, if one feature has high F-measure score (good discriminative power) but at the same time it is highly correlated (correlation > 0.9) to other variable with a higher F-measure score, if the model would include both features it would be producing similar outputs at cost of increasing complexity.

- **Stopping/selection criterion** (gray): this step of the algorithm aims to ensure that only features with both a good discriminative power and lower correlation are selected as explanatory/input variables of the ANN model while maintaining it as simple as possible. To this end, the relative weight X_f of each feature present in the "*Candidate_variables* array" when compared to each other is computed using the following expression: X_f= F_f/(sum(F_f). Then, only the features having an X_f value greater than 10% are selected to be used in the ANN model. It is important to note that without a stopping criterion of this kind, a feature with low discriminating power (i.e., that provides no added value to the model) but that shows low correlation with other variables would end up selected as explicative variable.

In Table 2 are presented the F-measure results (*F* score), as well as the final variables selected to be used as explanatory/input variables of ANN model (highlighted in bolt), after running the algorithm depicted in Figure 7 for the study case under analysis and for one of the disturbances selected (three phase symmetrical short circuit at bus 16). On the other hand, in Table 3 are presented the results of the correlation matrix for all the candidate variables that were analyzed within this process.

TABLE 2: FEATURE SELECTION ALGORITHM RESULTS: F-MEASURE SCORE AND FINAL FEATURES SELECTED TO BE USED AS INPUT OF THE ANN MODEL.

Features (f)	$\bar{x}_{f,secure}$	$\bar{x}_{f,insecure}$	$\sigma_{f,secure}$	$\sigma_{f,insecure}$	F-score (Ff)	F-score Rank	F-Score Criterion?	Correlation Criterion?	X(f) Criterion (X > 10 %)?	Final Selection
WFs_Total_Pgen	**1 277.39**	**2 278.14**	**697.21**	**183.38**	**1.14**	**1st**	**Selected**	**Not required to be assessed**	**Not required to be assessed**	**Selected**
Total_H_SMs	**33 971.69**	**26 734.08**	**2 440.76**	**5 710.53**	**0.89**	**2nd**	**Selected**	**Selected** (corr. < 0.9 with 1st feature selected)	**13.7%**	**Selected**
SGs_Total_Pgen	**3 109.26**	**1 781.36**	**682.51**	**901.04**	**0.84**	**3rd**	**Selected**	**Selected** (corr. < 0.9 with others already selected features)	**13.0%**	**Selected**
SG10_Bus30_Pgen	553.82	216.99	143.31	248.14	0.86	4th	Selected	Not Selected (corr. > 0.9 with others already selected features)	Not required to be assessed	Not selected
SG6_Bus35_Pgen	430.09	141.28	144.94	203.07	0.83	5th	Selected	Not Selected (corr. > 0.9 with others already selected features)	Not required to be assessed	Not selected
SG3_Bus32_Pgen	414.20	118.06	172.99	201.86	0.79	6th	Selected	Not Selected (corr. > 0.9 with others already selected features)	Not required to be assessed	Not selected
SG7_Bus36_Pgen	389.57	190.46	95.05	159.03	0.78	7th	Selected	Not Selected (corr. > 0.9 with others already selected features)	Not required to be assessed	Not selected
H_SC3_Bus38	**1 197.28**	**46.45**	**1 149.03**	**323.54**	**0.78**	**8th**	**Selected**	**Selected (corr. < 0.9 with others already selected features)**	**12.1%**	**Selected**
SG4_Bus33_Pgen	448.00	303.97	99.24	96.47	0.74	9th	Selected	Not Selected (corr. > 0.9 with others already selected features)	Not required to be assessed	Not selected
H_SC2_Bus37	**880.26**	**76.54**	**850.12**	**280.73**	**0.71**	**13th**	**Selected**	**Selected (corr. < 0.9 with others already selected features)**	**11.0%**	**Selected**
H_SC1_Bus34	**500.45**	**111.73**	**372.99**	**215.55**	**0.66**	**14th**	**Selected**	**Selected (corr. < 0.9 with others already selected features)**	**10.2%**	**Selected**
SG2_Bus31_Pgen	373.58	310.60	70.45	51.64	0.52	10th	Selected	Selected (corr. < 0.9 with others already selected features)	8.0%	Not selected
SG1_Bus39_Pgen	- 938.98	- 169.31	1 017.80	692.82	0.45	11th	Selected	Selected (corr. < 0.9 with others already selected features)	7.0%	Not selected
Load_Total_P	5 265.42	4 162.17	1 474.06	1 595.29	0.36	12th	Selected	Selected (corr. < 0.9 with others already selected features)	5.6%	Not selected
Total_SR	1 283.92	1 178.28	431.40	436.85	0.12	15th	Selected	Selected (corr. < 0.9 with others already selected features)	1.9%	Not selected

TABLE 3: FEATURE SELECTION ALGORITHM RESULTS: CORRELATION MATRIX.

Features	WFs_Total_Pgen	Total_H_SMs	SGs_Total_Pgen	SG10_Bus30_Pgen	SG6_Bus35_Pgen	SG3_Bus32_Pgen	SG7_Bus36_Pgen	H_SC3_Bus38	SG4_Bus33_Pgen	SG2_Bus31_Pgen	SG1_Bus39_Pgen	Load_Total_P	H_SC2_Bus37	H_SC1_Bus34	Total_SR
WFs Total Pgen	1.00	0.33	0.36	0.34	0.35	0.34	0.34	0.00	0.32	0.31	0.25	0.14	0.01	0.00	0.17
Total H SMs	0.33	1.00	0.81	0.81	0.85	0.80	0.77	0.01	0.65	0.48	0.36	0.48	0.01	0.00	0.06
SGs Total Pgen	0.36	0.81	**1.00**	**0.97**	**0.97**	**0.97**	**0.97**	0.00	**0.97**	0.67	0.66	0.76	0.00	0.00	0.64
SG10 Bus30 Pgen	0.34	0.81	**0.97**	**1.00**	**0.94**	**0.91**	**0.98**	0.00	**0.94**	0.57	0.67	0.76	0.00	0.00	0.58
SG6 Bus35 Pgen	0.35	0.85	**0.97**	**0.94**	**1.00**	**0.94**	**0.93**	0.00	**0.93**	0.59	0.63	0.73	0.00	0.00	0.54
SG3 Bus32 Pgen	0.34	0.80	**0.97**	**0.91**	**0.94**	**1.00**	**0.91**	0.00	**0.93**	0.62	0.62	0.72	0.00	0.00	0.59
SG7 Bus36 Pgen	0.34	0.77	**0.97**	**0.98**	**0.93**	**0.91**	**1.00**	0.00	**0.96**	0.57	0.69	0.77	0.00	0.00	0.63
H SC3 Bus38	0.00	0.01	0.00	0.00	0.00	0.00	0.00	**1.00**	0.00	0.02	0.00	0.00	0.01	0.00	0.01
SG4 Bus33 Pgen	0.32	0.65	**0.97**	**0.94**	**0.93**	**0.93**	**0.96**	0.00	**1.00**	0.59	0.73	0.81	0.01	0.00	0.79
SG2 Bus31 Pgen	0.31	0.48	0.67	0.57	0.59	0.62	0.57	0.02	0.59	**1.00**	0.26	0.34	0.02	0.00	0.50
SG1 Bus39 Pgen	0.25	0.36	0.66	0.67	0.63	0.62	0.69	0.00	0.73	0.26	**1.00**	0.86	0.02	0.00	0.64
Load Total P	0.14	0.48	0.76	0.76	0.73	0.72	0.77	0.00	0.81	0.34	0.86	**1.00**	0.01	0.00	0.65
H SC2 Bus37	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.02	0.02	0.01	**1.00**	0.00	0.02
H SC1 Bus34	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**1.00**	0.00
Total SR	0.17	0.06	0.64	0.58	0.54	0.59	0.63	0.01	0.79	0.50	0.64	0.65	0.02	0.00	**1.00**

3.1.3.2. *ANN Architecture and Training Process*

ANNs are a subset of machine learning techniques, namely, supervised learning – in the present case ANN are used as a regression algorithm, since the outputs (RoCoF and nadir) are continuous variables. It is noteworthy that however complex, it is possible to extract certain parameters from a trained ANN, such that it becomes possible to compute the outputs for a given set of inputs (explanatory variables). In Figure 8 is presented the general ANN architecture where the input variables were set after running the feature selection algorithm described in section 3.1.3.1 for the functional knowledge database generated in Block 2 (see section 3.1.2) – that respects to the New England test system chosen as study case.

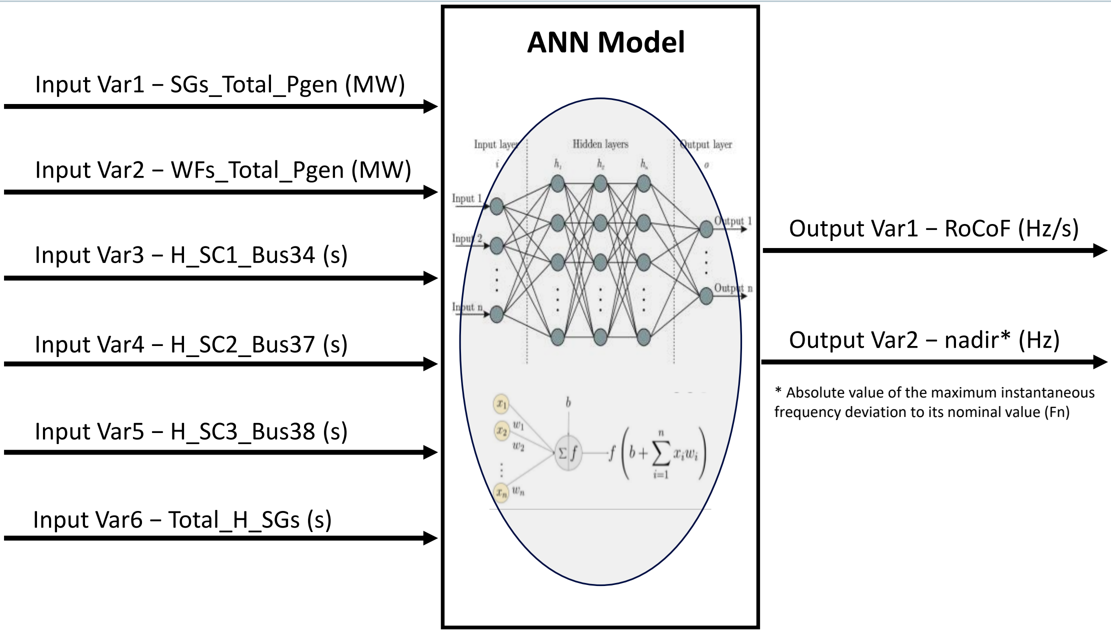


FIGURE 8: ANN GENERAL ARCHITECTURE/MODEL ADOPTED WITH THE FINAL INPUT/EXPLANATORY VARIABLES SELECTED FOR THE CASE STUDY UNDER ANALYSIS

As it can ben see in Figure 8, besides the definition of the input/output variables, the complete ANN model/architecture requires the definition of the number hidden layers, number of neurons of the hidden layer(s) and the action function of the hidden and output variables. With the exception of the output variables[9], the optimal values of the remaining aspects depend on the nature of the problem, thus there is no rule of thumb to tune these parameters although some guidelines are provided in literature [28]–[31]. Therefore, a proper definition of the parameters relies on a trial-and-error approach for most of the problems, and on the analysis of the Root Mean Square Error (RMSE) and Mean Square Error (MAE) of the training and test data set. For the study case presented, several preliminary tests were conducted to find the most suitable values for these parameters through the analysis of the referred indicators. At the end, the ANN architecture that have led to the best results, in terms of estimation accuracy and computational time, is as follows: one hidden layer with 12 neurons, using a sigmoid activation function in the hidden layer and a linear function activation in the output layer.

The training process aims to produce a properly trained MLA structure (to be used as a “black Box”) capable of being used online, in real time, to accurately assess dynamic security with respect to frequency stability. Hence, the training process will allow MLA to learn the dynamic behavior of the system from the functional knowledge generated in the functional block 2 as described in section 3.1.2.

It is important to note that the training process of the ANN was based on the Adam optimization algorithm [32], which consists in a stochastic gradient descent method. Therefore, all the tests were carried out using this training method that was implemented by means of the TensorFlow Python library (Keras API). The Keras API provides at the end an encrypted ANN trained object (file with .h5 extension) that can be loaded and run afterwards. Regarding the stop training criterion, it was based in the *Early Stopping* technique, which tries to avoid overfitting. This technique stops the learning process when the loss function does not change over a series of epochs according with a pre-defined argument/parameter named “patience” in Keras library. For example, if we want the validation

[9] Taking into account the scope and the objectives of the DSA tool, the output variables were assumed from the beginning to be necessarily RoCoF and nadir.

accuracy to increase, and the algorithm to stop if it does not increase for 20 periods, the "patience" parameter should be set to 20. In fact, this was the value adopted in the scope of the training process of the DSA tool for the New England test case under analysis. Further details about this library and about the execution process of the ANN are provided in section 3.1.4.1 (functional Block 4).

Still regarding the training process, the global database of functional knowledge generated in the functional block 2 (see section 3.1.2) needs to be firstly converted into a smaller dataset, suitable to be used directly in the training and testing processes. This task was done by separating, for each disturbance, the data of the input and output variables of all OPs. Then, 70% of the generated dataset was used for training purposes, while 15% was used for validation and 15% for testing (see section 3.1.4.4). The metric to evaluate the ANNs training process was both the RMSE and MAE. Both these metrics were also used to evaluate the performance for the test set.

Moreover, in order to equally distribute importance of each input variable, and thus improve the performance and training stability of the model, input data was normalized using the Min-Max normalization method. In this method, for every feature, the minimum value of that feature gets transformed into a 0, the maximum value gets transformed into a 1, and every other value gets transformed into a decimal between 0 and 1.

Another aspect that deserves attention is that each disturbance must be dealt with a proper ANN. This is required to avoid having conflicting data that may disrupt the learning process ANN, which would be the case if only one ANN was trained for all contingencies. On contrary, having one ANN trained for each disturbance, it is produced a properly trained MLA structure (to be used as a "black Box") capable of being used online, in real time, or offline to accurately assess dynamic security with respect to frequency stability. Hence, recalling that for the New England test system under study a set of 3 contingencies were selected as being critical to the system, a total of 3 different ANN were required to be trained. During the training process, each ANN was trained using as input data only information from the functional knowledge database related to the correspondent contingency.

3.1.4.Internal Block 4 – Tool Execution Process

The execution of the tool is performed in the functional block 4 (see Figure 1 in section 1) and can be seen as the "core" procedure of DSA methodology envisioned. In general terms, it corresponds to run N already trained ANN (one per contingency up to N contingencies) for a single OP or for several OPs. More specifically, each ANN object (.h5 file) is loaded with the values of the explicative/input variables correspondent to the OP(s) to be analyzed and then run. At the end of the process, the RoCoF and nadir estimated values are provided as outputs together with information about the SC that need to be turned on to bring the system to a secure state whenever the system is classified as unsecure.

The execution of the tool can be better understood observing the diagram depicted in Figure 9, which illustrates the whole process.

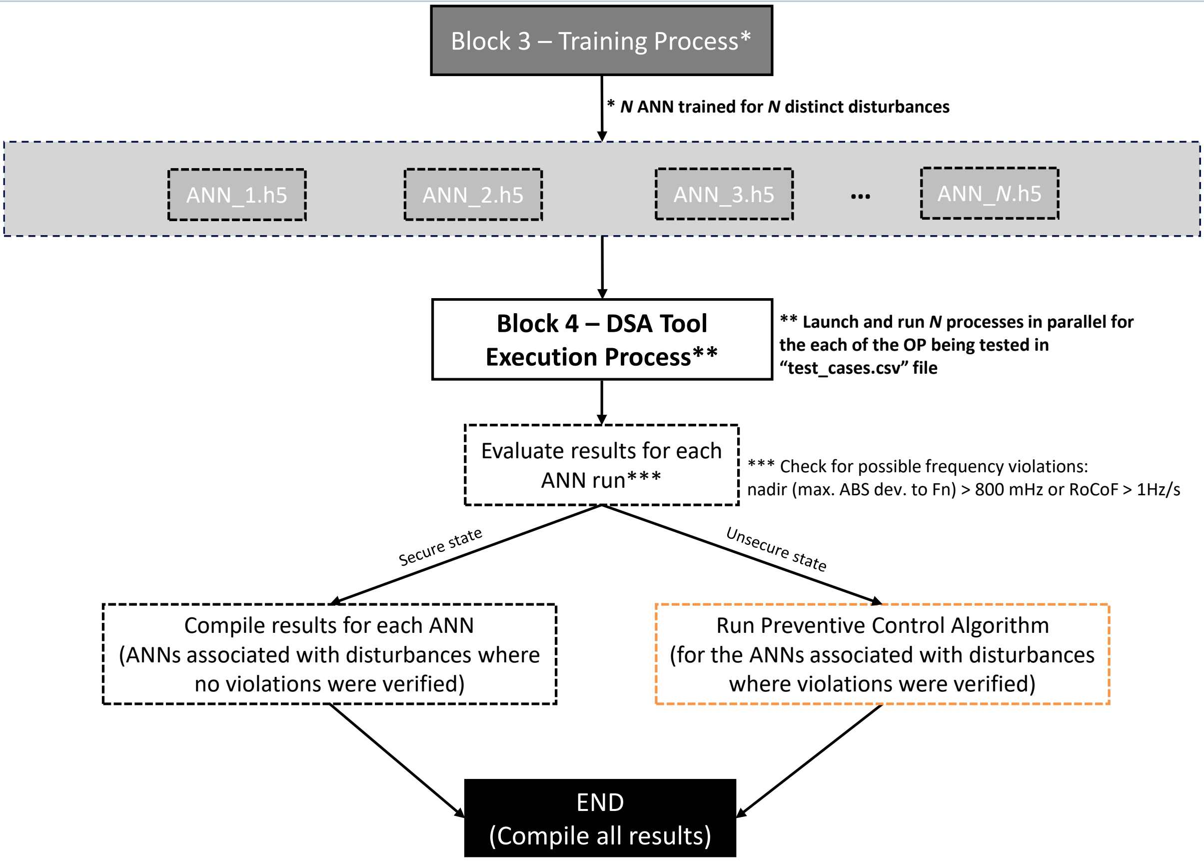


FIGURE 9: OVERVIEW OF THE EXECUTION PROCESS OF THE TOOL FOR N ALREADY TRAINED ANN

The preventive control algorithm (orange square in Figure 9) is run through a function within the execution process of the DSA tool. It aims at inferring/suggesting possible measures that may be taken to ensure that the system is secure from a dynamic point of view (regarding the frequency stability indicators nadir and RoCoF). In other words, the algorithm indicates the SC that need to be dispatched (based on a pre-established merit order[10]) to bring the system to a secure state (whenever the system is classified as unsecure). The proposed algorithm is detailed in the flowchart presented in Figure 10.

[10] For the New England test system, the merit order for the SC(s) that are turned on firstly was defined according to the following sequence of elements *(SC_Merit_Order_List* in Figure 10): 1st – [SC1], 2nd – [SC2], 3rd – [SC3], 4th [SC1, SC2], 5th [SC1, SC3], 6th [SC2, SC3], 7th [SC1, SC2, SC3] (see Figure 2 in section 2.1).

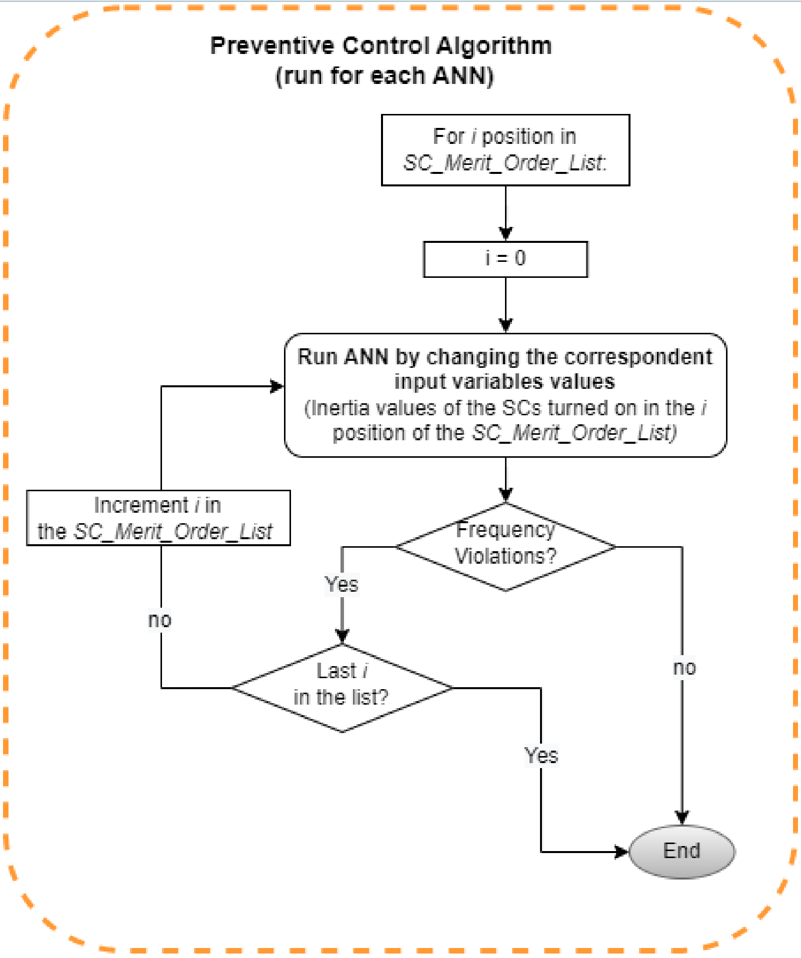


FIGURE 10: PREVENTIVE CONTROL ALGORITHM DEVELOPED TO INFER/SUGGEST POSSIBLE MEASURES THAT ARE CAPABLE OF MAINTAINING SYSTEM SECURITY (BASED ON BRINING ONLINE TO THE SYSTEM SC – WHENEVER AN UNSECURE STATE IS VERIFIED)

Even though the tool runs usually fast even for several ANN, as it can be seen in Figure 9, in order to ensure that the tool is fit for online applications, it was designed to launch N ANN processes in parallel, being the data synchronized and compiled at the end.

3.1.4.1. Hardware and Software Requirements

The execution of the tool is not a time-consuming process, thus not requiring a significant amount of data storage (ca. 8 GB are more than enough) nor a very high processing power. In fact, the simulations performed over the IEEE 39-bus system selected as study case have run in less than a second in a common laptop available in the market nowadays (see main characteristics in Figure 11). Even though that this task is mainly intended to be executed online[11] at the SCADA/EMS level (either considering its standalone version or integrated version with T4.5 tool) the computational requirements are minimal.

[11] The DSA tool was also designed for off-line purposes (e.g., operational planning). In such cases, it can be used for instance to assess dynamic security of day ahead market solutions either considering its standalone version or integrated with ATTEST T4.4 tool (see section 4).

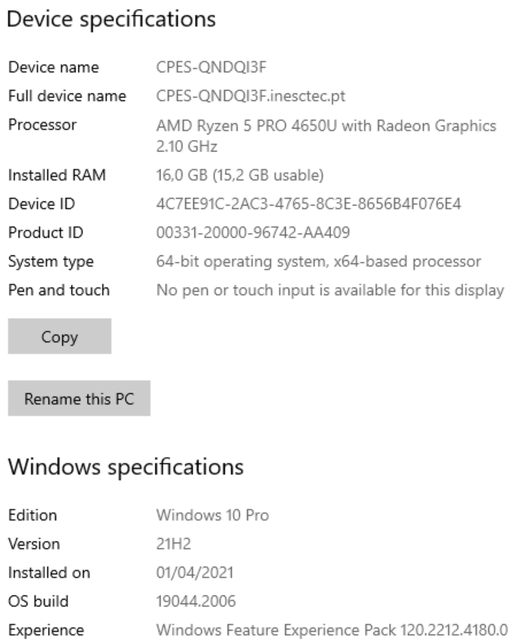


FIGURE 11: HARDWARE AND WINDOWS OS SPECIFICATIONS OF THE NOTEBOOK USED TO RUN THE TOOL

In terms of software requirements, this tool requires the version 3.7 or higher of Python (Python 3.9 was used to developed and test the tool) for Windows OS with the following additional Python packages/libraires installed:

- NumPy (1.19.5 release): used to perform a wide variety of mathematical operations on arrays. In addition, it also provides functions for working in domain of linear algebra, Fourier transform, and matrices by powerful data structures to Python that guarantee efficient calculations. It is an open-source project that can be installed and used freely.
- TensorFlow (2.5.0 release): is a Python-friendly open-source library for numerical computation, providing a strong support for machine learning, deep learning and ANNs, making its developing and learning faster and easier.
- Keras API (provided within the TensorFlow library): is a high-level, deep learning API developed for implementing ANNs. It allows to make the implementation of neural networks easy. It also supports multiple backend neural network computation.

3.1.4.2. Mathematical model

In order to enable the integration of the DSA tool with other ATTEST tools, namely with tools T4.4. and T4.5 (see section 4), the mathematical model that corresponds to the tool execution needs to be formulated and described.

As it was already mentioned, the execution process of the tool corresponds to load an already trained ANN with defined input variables (the same used for training the ANN) and then run the ANN model that will compute the outputs for a given set of inputs values. This process was implemented making use of the Keras API functions (provided within TensorFLow Python library). More specifically, this is done by calling Keras API method "model.predict()". The corresponding mathematical model that is implemented (and runs) inside the "predict()" method is explained bellow.

The parameters that need to be extracted from a trained ANN in order to compute the outputs as a function of inputs are the following:

- Weights matrix, *[WIH]*, with dimension *i x h*, where *i* is the number of inputs (features) and *h* is the number of neurons in the hidden layer,
- Bias array, *[BH]*, with dimension *h x 1*,
- Weights matrix, *[WHO]*, with dimension *h x o*, where *o* is the number of outputs (o=2, RoCoF and nadir in this case),
- Bias array, *[BO]*, with dimension *o x 1,*
- Maximum and minimum values for each explanatory variable in the training set.

The steps for computing the outputs based on a set of inputs are as follows:

1. Transpose *[WIH]* (after which dimension becomes *h x i*),
2. Declare the input array *[I]*, with dimension *i x 1*, which contains the values of explanatory variables
    a. Normalize each value *x* of the array using the expression, in relation to the maximum and minimum values for each explicative variable in the training set:

$$x_{normalized} = \frac{(x - x_{min})}{(x_{max} - x_{min})}$$

3. Obtain *[VH]*, the product of *[WIH].[I]* which is an array with dimension *12 x 1*
4. Obtain *[H] = [VH] + [BH],* an array with dimension *12 x 1*
5. Obtain *[AH]* by running the sigmoid activation function *S(x)* on each element *x* of the array [*H*]:

$$S(x) = \frac{1}{1 + e^{-x}}$$

6. Transpose *[WHO]* (after which dimension becomes *o x h* or *2 x 12*),
7. Obtain *[VO],* the product of *[WHO].[AH]*, which is an array with dimension *2 x 1*
8. Obtain *[O] = [VO] + [BO],* an array with dimension *2 x 1*
    a. Since a linear function is used on the output, there is no need to run any further activation function for each element of the array *[O]*
9. De-normalize the values of *[O]*

$$x_{denormalized} = x_{min} + x.(x_{max} - x_{min})$$

3.1.4.3. Interface and User Guide for Running the Tool

The tool can be executed from a command line prompt by running the corresponding Python script through the command "Running_DSA_tool_main.py"[12] without any arguments being passed. Note that the input arguments required have been coded internally inside the tool and are read and passed to it through external files (more details are presented further in this section). Instructions regarding the tool execution are given in case any error occurs during its execution, namely if the arguments (input files) are incorrect or insufficient/inconsistent data is detected. In this sense, the following code errors (non-related with Python exceptions) have been coded and are returned by the tool:

- Code 0 _– "Process finished with no errors. The tool run successfully for *N* ANN (*N* Contingencies) and *X* OPs", where *N* is replaced by the number of ANN/contingencies that were run in parallel by the tool and *X* replaced by the number of OPs tested;

[12] Basic Python IDLE that comes included in Python installations, or other more advanced Integrated Development Environments (IDEs) such as Pycharm, Spider, among several others that are free available and can be used to run the tool.

- Code 1 – "Error: "ANN_id.h5" file is missing. Please verify its name and ensure that its path corresponds to the directory where tool Python script is located".
- Code 2 – "Error: "Config.csv" file is missing. Please verify its name and ensure that its path corresponds to the directory where tool Python script is located".
- Code 3 – "Error: "Test_cases.csv" file is missing. Please verify its name and ensure that its path corresponds to the directory where tool Python script is located".
- Code 4 – "Error: Bad, insufficient or inconsistent data detected in "Config.csv" file. Please revise the whole file carefully including data format and values".
- Code 5 – "Error: Bad, insufficient or inconsistent data detected in "Test_cases.csv" file. Please revise the whole file carefully including data format and values".
- Code 6 – "An unexpected error occurred. Please contact the tool developer".

After the execution process is finished, in addition to the error codes returned, it is displayed information about the total running time ("Elapsed time") and a message informing the user to check the "results.csv" file where all the results are compiled and can be properly analyzed. In Figure 12 is shown an illustrative example of the DSA tool execution through the command line interface.

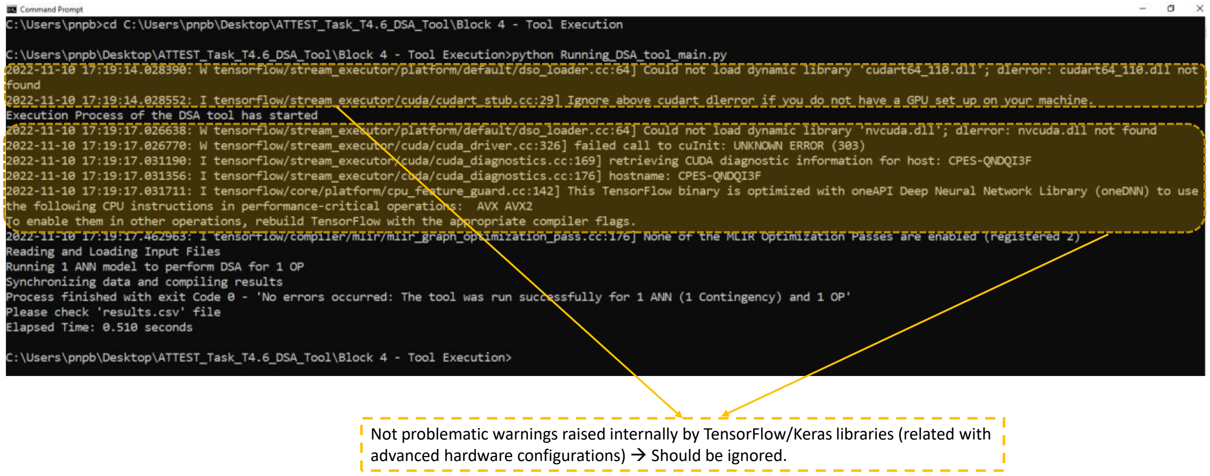


FIGURE 12 - DSA TOOL EXECUTION. COMMAND LINE INTERFACE.

Internally, the tool receives and processes at least three arguments[13] that corresponds to the following files:

- "ANN_id.h5" – this file is an encrypted file (TensorFlow/Keras object) that contains the model of an ANN already trained for given contingency (see section 3.1.3.2).
- "config.csv" – contains information related with control parameters of the tool and data required for the standardization method, which is run during the tool execution. In particular, the following data is included in this file: number of the OPs to be run/ tested, number of ANN to be run, number of ANN inputs variables, number of ANN output variables, and maximum and minimum values of each ANN input variable (gathered from the training dataset) to be used by the standardization method.
- "test_cases.csv" – this file contains the values of the variables selected to be used as input of the ANN model correspondent to the OPs to be tested.

[13] Note that in the file "ANN_id.h5", the "id" is set according to the number *N* of ANNs trained, which should be equal to the number of critical contingencies simulated (see Figure 9). Hence, the number of "ANN_id.h5" files to be passed as argument may vary between 1 and *N*.

In Figure 13 and Figure 14 is shown, respectively for the "config.csv" and for "test_cases.csv" files, an illustrative example of the data format proposed. For a better reading, both files were opened with Excel converting commas to tabs. The files were filled with data relative to the New England test system under analysis. As it can be seen, in this case, the tool will assess dynamic security for 1 OP/test case through 1 ANN model (one critical contingency is evaluated) with 6 input variables and 2 output variables (see Figure 8 in section 3.1.3.2).

	A	B	C	D	E	F	G	H	I
7	/Nr. of ANN Input Vars								
8	6								
9									
10	/Nr. of ANN Output Vars								
11	2								
12									
13	/Min. and Max. values of the ANN Vars to be used by the Standardization method during the running process of the tool								
14	/ Data gathered from the training dataset								
15				Min. Value	Max. Value				
16	ANN1	Input Var1	SGs_Total_Pgen (MW)	1000.72	3842.75				
17	ANN1	Input Var2	WFs_Total_Pgen (MW)	0.00	2500.00				
18	ANN1	Input Var3	H_SC1_Bus34 (s)	0.00	866.40				
19	ANN1	Input Var4	H_SC2_Bus37 (s)	0.00	1619.80				
20	ANN1	Input Var5	H_SC3_Bus38 (s)	0.00	2300.00				
21	ANN1	Input Var6	Total_H_SMs (s)	20890.30	34790.00				
22	ANN1	Output Var1	RoCoF (Hz/s)	0.16	1.84				
23	ANN1	Output Var2	nadir (max. deviation to Fn) (	0.06	1.02				
24	/...	/...	/...	/...	/...				
25	/Fill in next rows using the same formatting if more ANNs are considered								

FIGURE 13 – "CONFIG.CSV" FILE DATA FORMAT PROPOSED. ILLUSTRATIVE EXAMPLE FOR THE NEW ENGLAND TEST SYSTEM ADMITTING 1 ANN MODEL TRAINED FOR 1 CONTINGENCY WITH 6 INPUT VARIABLES AND 2 OUTPUTS VARIABLES.

	A	B	C	D	E	F	G
1	Test Case Id	Input Var1	Input Var2	Input Var3	Input Var4	Input Var5	Input Var6
2	1	1037.34	2407.80	0.00	0.00	0.00	20890.30
3	/...	/...	/...	/...	/...	/...	/...
4	/Fill in next rows using the same formatting if more OPs are considered						

FIGURE 14 – "TEST_CASES.CSV" FILE DATA FORMAT PROPOSED. ILLUSTRATIVE EXAMPLE FOR THE NEW ENGLAND TEST SYSTEM WITH DATA FOR 1 OP / TEST CASE

In addition to the input files required for executing the tool, a complementary informative Excel file (complementary_information.xlsx") is also provided. This file contains information related with functional Block 1, Block 2, and Block 3, namely about the characteristics of test system, the critical contingencies/disturbances simulated, the ANN model/architecture, and about the training process of the tool, among other information that can be of interest to the user / TSO. In Figure 16 it is presented an example of the "complementary_information.xlsx" file for the New England test system selected as case study.

	A	B	C
1	DSA Tool - Test Sytem Characterization		
2	Name:	"10 machine New-England Power System"modified to meet the purpose of the ATTEST T4.6 Task (DSA Tool)	
3	Transmission System Nominal Voltages	345 kV, 230 kV & 138 kV	
4	Fn (Nominal Frequency)	60 Hz	
5	Nr. Of Buses	42	
6	Nr. Loads	31	
7	Nr. Lines	34	
8	Nr. Of 2 Windings Transf.	15 (3 Transmssion transf. and 12 generators step up transf.)	
9	Nr. Of SGs	7	
10	Nr. Of SCs	3	Total of 2500 MVA: SC1 - 800 MVA; SC2 -700 MVA; SC3 - 1000 MVA
11	RES	3 WFs	Total of 2500 MW: WF1 - 800 MW; WF2 - 700 MW; WF3 - 1000 MW
12			
13			
14	DSA Tool - Internal Block 1 (OPs Generation and Critical Disturbances Identification)		
15	Nr. OPs Generated	534000	
16	Load Range (OPs)	[2700, 8060] MW --> OPs generated in 20 MW steps	
17	WFs Active Power Range (OPs)	[0, 2500] MW --> OPs generated in 20 MW steps	
18	Critical Contingencies/Disturbances identified:		
19	Contingency id 1	three phase symmetrical short circuit at bus 16 (Zf≈0 and fault clearing time = 250 ms)	
20	Contingency id 2	three phase symmetrical short circuit at bus 31 (Zf≈0 and fault clearing time = 250 ms)	
21	Contingency id 3	three phase symmetrical short circuit at bus 2 (Zf≈0 and fault clearing time = 250 ms)	
22			
23			
24	DSA Tool - Internal Block 2 (Generation of functional knowledge through time-domain dynamic simulations)		
25	In average 85% of OPs are likely to be secure for all the disturbances analysed		
26	In average 15% of OPs might be unsecure (system security may be at at risk regarding dynamic frequency stability (violations in freq. nadir and/or RoCOF regulated values)		
27			
28			
29	DSA Tool - Internal Block 3 (ANN Model/Architecture and Training Process)		
30	ANN model tyoe	Feed-forward ANN- Sequential model (linear stack of layers)	
31	Nr. of hidden ayers	1	
32	Nr. Of hidden neurons	12	
33	Activation Function (Hidden layer)	Sigmoid	
34	Activation Function (Output layer)	Linear	
35	Training Algorithm	Adam optimization algorithm using Early Stopping Criterion to avoid overfitting("patiente" parameter set to 20)	
36	Standardization/Normalization Process	Normalization between [0,1] using Min-Max Method	
37	Implementation of the ANN model and of the training algorithm	Python TensorFlow package (Keras Library)	
38	Training set	373800	
39	Validation set	80100	
40	Test set	80100	

FIGURE 15 – "COMPLEMENTARY_INFORMATION.XLSX" FILE DATA FORMAT PROPOSED. ILLUSTRATIVE EXAMPLE FOR THE NEW ENGLAND TEST SYSTEM

It should be noted that all the input files ("ANN_id.h5", "config.csv" and "test_cases.csv") must be in the same directory of where the tool script "Running_DSA_tool_main.py" is located. Figure 16 shows an example of a proper directory structure that could be used for running the tool.

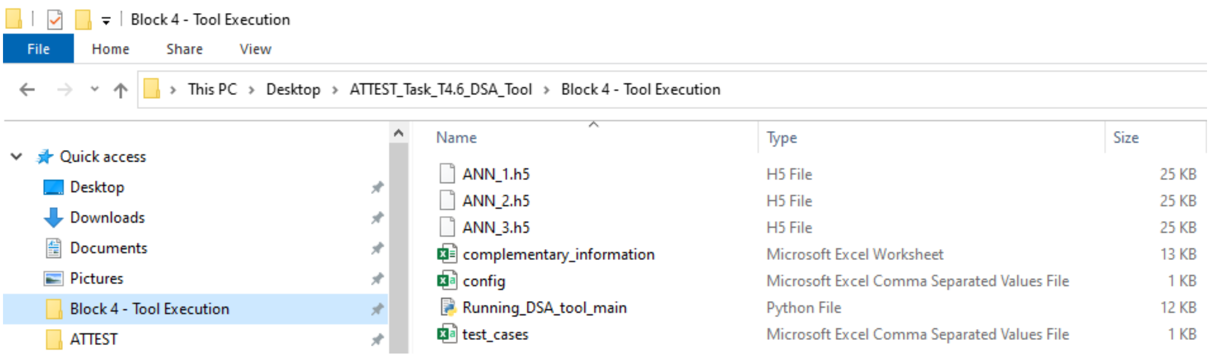


FIGURE 16 –ILLUSTRATIVE EXAMPLE OF A PROPER DIRECTORY STRUCTURE FOR RUNNING THE DSA TOOL

Figure 17 presents an example of the "results.csv" file, generated after running the tool for the New England test system. In this case, the tool was run using one ANN trained for one critical contingency (model loaded from the ANN_1.h5 file), being one critical OP tested according to the data defined in "test_cases.csv" file. Regarding the "config.csv" file, it used the data provided in the example of Figure 13. The results for this case show that the system security may be at risk due to violations of the defined limits both in nadir and RoCof, being the contingency with id 1 the most critical. Thus, the tool suggests that 3 SC should be turned on (SC1, SC2 and SC3) in order to bring the system to a secure region.

	A	B	C	D	E	F	G
1	ANN / Contingency / Disturbance Id	Estimated RoCoF (Hz/s)	Estimated nadir (deviation to the Nominal Freq.) (Hz)	OP Secure?	SC(s) needed to ensure security for one contingency (see Id)	Estimated RoCoF (Hz/s) with SCs	Estimated nadir with SC(s) (deviation to the Nominal Freq.) (Hz)
2	1	1.7876084	0.9771807	No	SC3_bus38	0.95518804	0.56678855
3	/...	/...	/...	/...	/...	/...	/...
4	/Next rows will have results in the same formatting if more Ops (test cases) are considered in the "test_cases.csv" file						
5							
6	SC(s) needed to ensure system security for all contingencies /disturbances						
7	SC3_bus38						

FIGURE 17 – "RESULTS.CSV" FILE DATA FORMAT. ILLUSTRATIVE EXAMPLE AFTER RUNNING THE TOOL FOR ONE TEST CASE CORRESPONDENT TO A CRITICAL OP (I.E., LOW CONSUMPTION AND VERY HIGH RES LEVELS PRESENT IN THE SYSTEM, THUS MEANING A LOW SYNCHRONOUS INERTIA)

It important to mention that when the tool is run online it is expected that a single OP is evaluated at a time, whereas when running it off-line (e.g., for day-ahead operational planning purposes), several OPs are expected to be assessed at a time.

Finally, it is worth mentioning that a proper graphical user interface (GUI) for this tool is envisioned to be further developed by the partners in the scope of WP6.

3.1.4.4. Performance Evaluation – Results and Illustrative Examples

In order to evaluate the tool performance, the DSA tool was run for a Testing Set (TS) that corresponds to 15% of the total OPs generated in Block 1 (see section 3.1.1.2), meaning that a total of 80100 OPs were evaluated. It should be recalled that this dataset is different from the one used for training and validation purposes. The test dataset was randomly generated based on the data gathered from the functional knowledge database created in Block 2 (see section 3.1.2) after being processed by a feature selection method in the scope of Block 3 (see section 3.1.3.1).

The performance of the DSA tool was evaluated for the TS by comparing the real nadir and RoCoF values that resulted from the PSS/E dynamic simulations performed in Block 2 (see section 3.1.2) with the corresponding ones that were estimated by the DSA tool (tool's outputs) after its execution, and then analyzing them considering the following aspects: accuracy/quality, comprehensibility, classification errors, and computational efficiency. In particular, the following numerical indices were computed:

Mean Absolute Error (MAE) given by:

$$MAE = \frac{1}{N(TS)} \sum_{OP_i \,\epsilon\, TS} |y_i - \hat{y}_i(OP_i)|$$

Root Mean Squared Error (RMSE) given by:

$$RMSE = \sqrt{\frac{1}{N(TS)} \sum_{OP_i \,\epsilon\, TS} \left(y_i - \hat{y}_i(OP_i)\right)^2}$$

where:

- $N(TS)$: Number of operating points (OPs) in the Testing Set (*TS*);
- y_i: Real value of the security index, for OP_i;
- $\hat{y}_i$: Value estimated by the ANN structure, for the security index of OP_i.

The classification accuracy can be inferred by the following misclassifications rates:

$$Global\ Class.Error = \frac{Nº\{OPs\ of\ the\ TS\ incorrectly\ class.\}}{Nº\{OPs\ of\ the\ TS\}} \times 100\%$$

$$False\ Alarm\ Error = \frac{Nº\{"secure"\ OPs\ of\ the\ TS\ class.\ as\ "unsecure"\}}{Nº\{"secure"\ OPs\ of\ the\ TS\ \}} \times 100\%$$

$$Missed\ Alarm\ Error = \frac{Nº\{"unsecure"\ OPs\ of\ the\ TS\ class.\ as\ "secure"\}}{Nº\{"unsecure"\ OPs\ of\ the\ TS\ \}} \times 100\%$$

Regarding the computational efficiency of the tool, it should be noted that each test case analyzed during the testing process can be seen as assessing the dynamic security in a given operating scenario (one snapshot). Therefore, while the computational efficiency for the training process may be assessed by calculating the total time required, for the testing process it was considered the maximum value observed for the total number of TS evaluated, after running the tool individually for each case.

Figure 18 and Figure 19 compare respectively the results estimated by the DSA tool for frequency nadir (maximum absolute deviation to Fn) and for RoCof with the corresponding real values obtained from the dynamic simulation performed in PSS/E in the scope of Block 2 (see section 3.1.2.) for all the OPs in the TS analyzed. As it can be seen in these figures, although there are some outliers (a very small percentage), there is a strong correlation (almost linear) between the estimated and the real values, both for nadir and RoCoF. These results evidence the good performance of the tool regarding its accuracy on estimating these indicators.

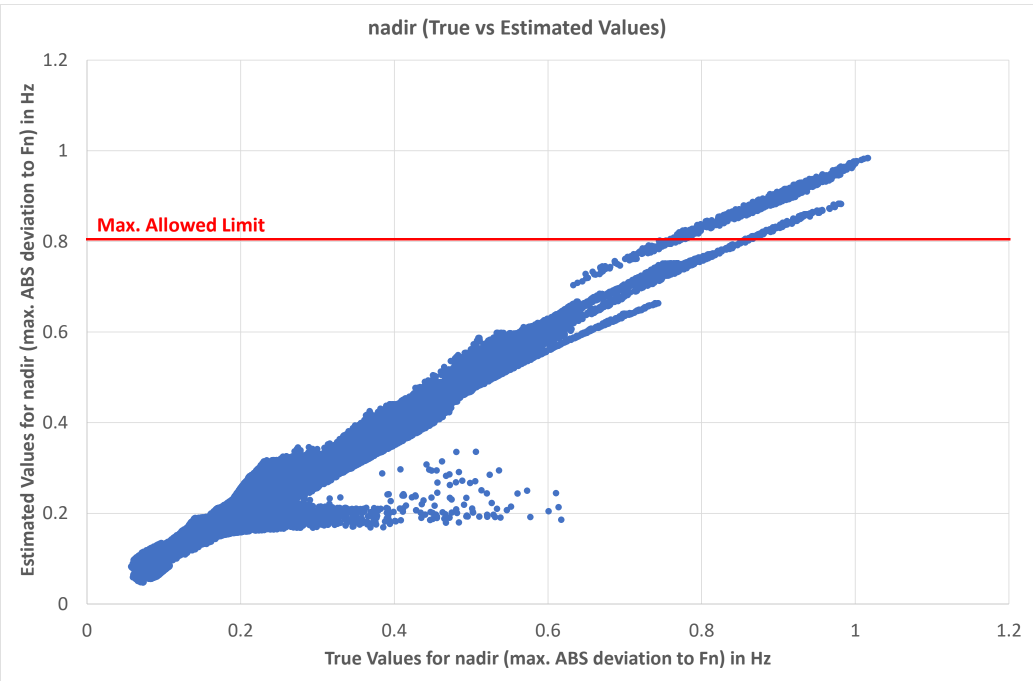


FIGURE 18 – COMPARISON BETWEEN TRUE AND THE ESTIMATED VALUES FOR NADIR ATTAINED AFTER RUNNING THE TOOL FOR ALL THE OPS IN THE TS (CONSIDERING THE CRITICAL CONTINGENCY WITH ID 1 – THE MOST CRITICAL).

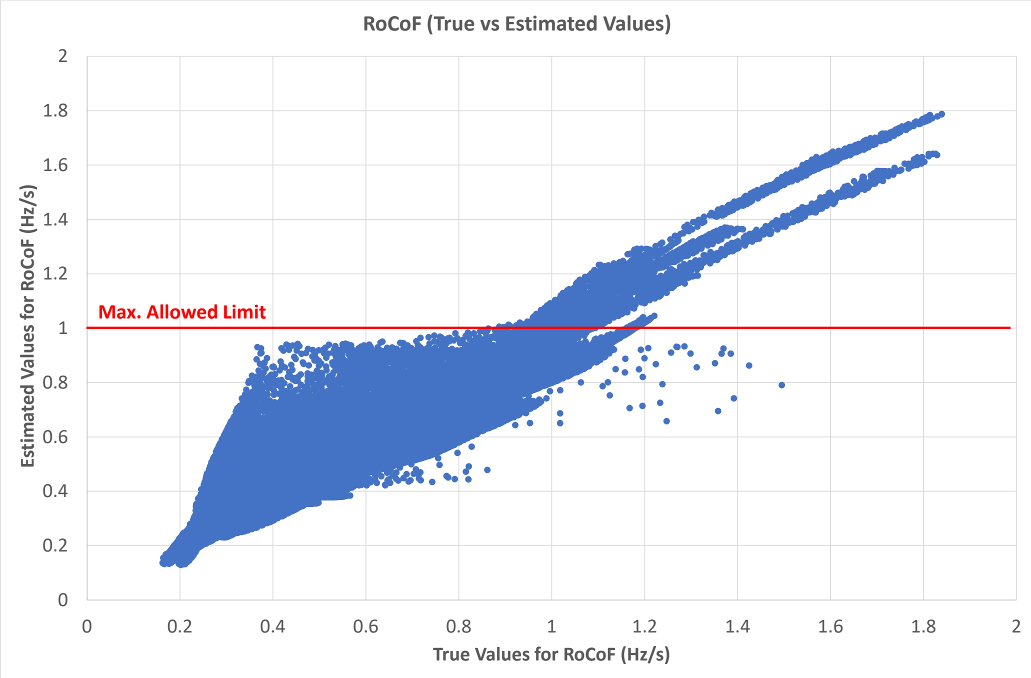


FIGURE 19 – COMPARISON BETWEEN TRUE AND THE ESTIMATED VALUES FOR ROCOF ATTAINED AFTER RUNNING THE TOOL FOR ALL THE OPS IN THE TS (CONSIDERING THE CRITICAL CONTINGENCY WITH ID 1 – THE MOST CRITICAL).

In Table 4 are compiled the results attained for all the performance indexes that were characterized at the beginning of the current section. These results respect to all the cases analysed in the test dataset assumed for the study case of New England test system and to the most critical contingency identified (contingency 1). Analysing the results obtained, it can be concluded that the tool accounts for very promising results for both estimation accuracy and computational efficiency regarding all the numerical indicators analyzed. For instance, in terms of the global classification error, around 99.11% of the cases were correctly classified regarding RoCof indicator and 100% regarding nadir. Moreover, only 0,29% of the cases are badly classified regarding false alarms index (system is on an unsecured state but was estimated as being secure), being this index probably the most relevant indicator for the TSO. When it comes to the computational efficiency (running time performance), the maximum elapsed time registered was below 1 second, thus confirming that the tool is suitable for online applications.

TABLE 4: COMPILED RESULTS FOR ALL THE PERFORMANCE INDICATORS ANALYZED. RESULTS RESPECT TO ALL THE CASES OF THE TEST DATASET AND TO THE CRITICAL CONTINGENCY WITH ID 1(THE MOST CRITICAL).

	Tool's Performance					
Accuracy Performance						
Performance Indexes	**RoCoF**			**nadir (absolute value deviation to Fn)**		
Classification	**Global Classification Error**	**Missed Alarms ("Unsecure" is true but "Secure" is Estimated - most important index)**	False Alarms ("Secure" is true but Unsecure is "Estimated")	**Global Classification Error**	**Missed Alarms ("Unsecure" is true but "Secure" is Estimated - most important index)**	False Alarms ("Secure" is true but Unsecure is "Estimated")
	99.11% accuracy	**0.72 % badly estimated (99.28% accuracy)**	0.29 % badly estimated (99.71% accuracy)	**100% accuracy**	**100% accuracy**	100% accuracy
MAE	**45.24 mHz / s**			**10.91 mHz**		
RMSE	66.36 mHz / s			17.4 mHz		
Time Performance						
Running Time	<1s (maximum elapsed time obtained when running the tool for all the OPs in TS individually)					

4. Integration/Interactions with other ATTEST Tools

The DSA tool (Task T4.6) addressed in this document was envisioned to interact with ATTEST's tools T4.4. and T4.5 as depicted in the diagram of Figure 20 taken from Deliverable D4.1 [1].

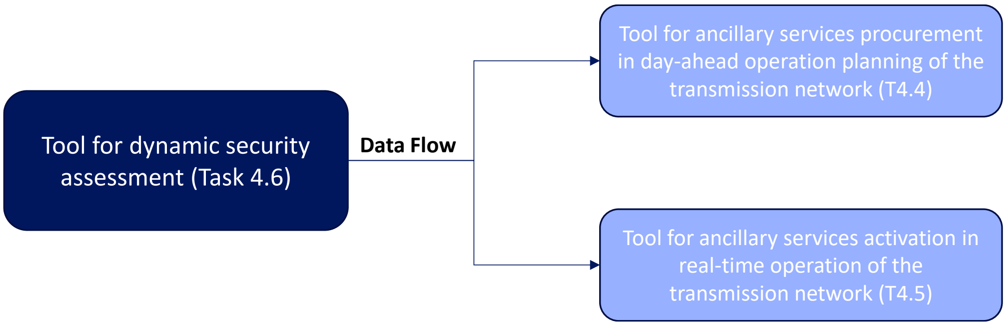


FIGURE 20: INTERACTION OF THE TOOL FOR DSA T4.6 WITH OTHER ATTEST TOOLS

In general terms, the integration process of the tools consisted on the inclusion of the mathematical model correspondent to the execution process of the DSA tool (see section 3.1.4.2) in both the AC security-constrained optimal power flow (SCOPF) formulation for T4.4 tool and in the AC OPF formulation for T4.5 tool. In this process, some of the explicate variables of the ANN were modeled as decision variables in the corresponding objective functions of both tools, whereas others were modeled as independent variables. More specifically, the variables *SGs_Total_Pgen* (Input Var1), *WFs_Total_Pgen* (Input Var2), *H_SC1_Bus34* (Input Var3), H_*SC2_Bus37* (Input Var4) and H_*SC3_Bus38* (Input Var5) were model as decision variables, while the remaining one (*Total_H_SGs* – Input Var6) was modelled as independent variable (see Figure 8 in section 3.1.3.2).

In Figure 21 and Figure 22 it is presented a detailed flowchart with the envisioned interactions of the DSA tool when integrated with the T4.4 and T4.5 tools.

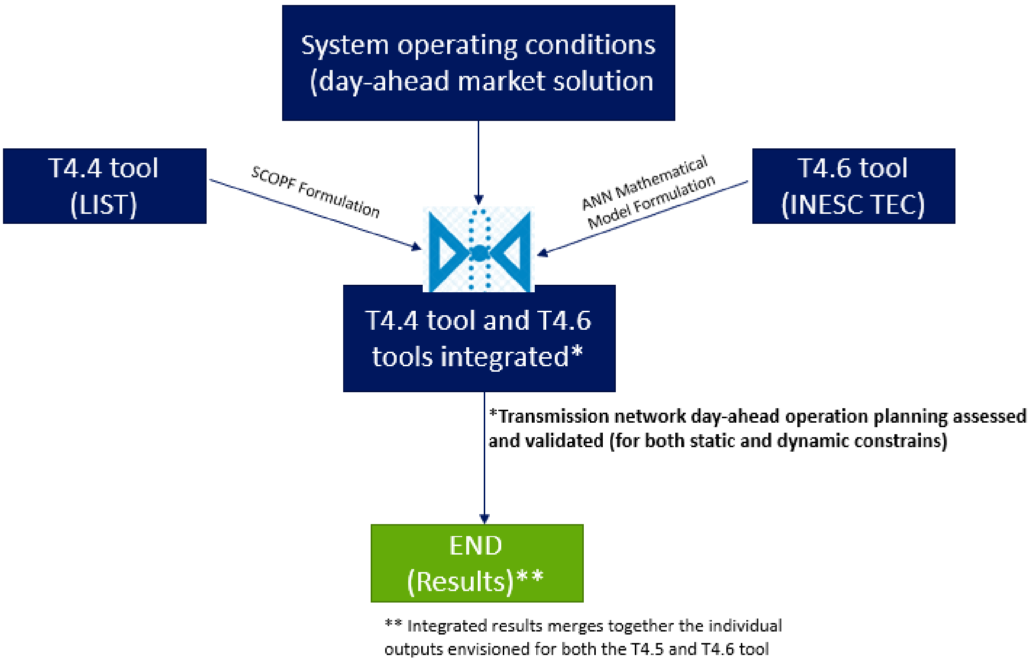


FIGURE 21: INTERACTIONS OF THE TOOL FOR DSA T4.6 WITH THE T4.4 TOOL CONSIDERING AN INTEGRATED VERSION OF BOTH TOOLS

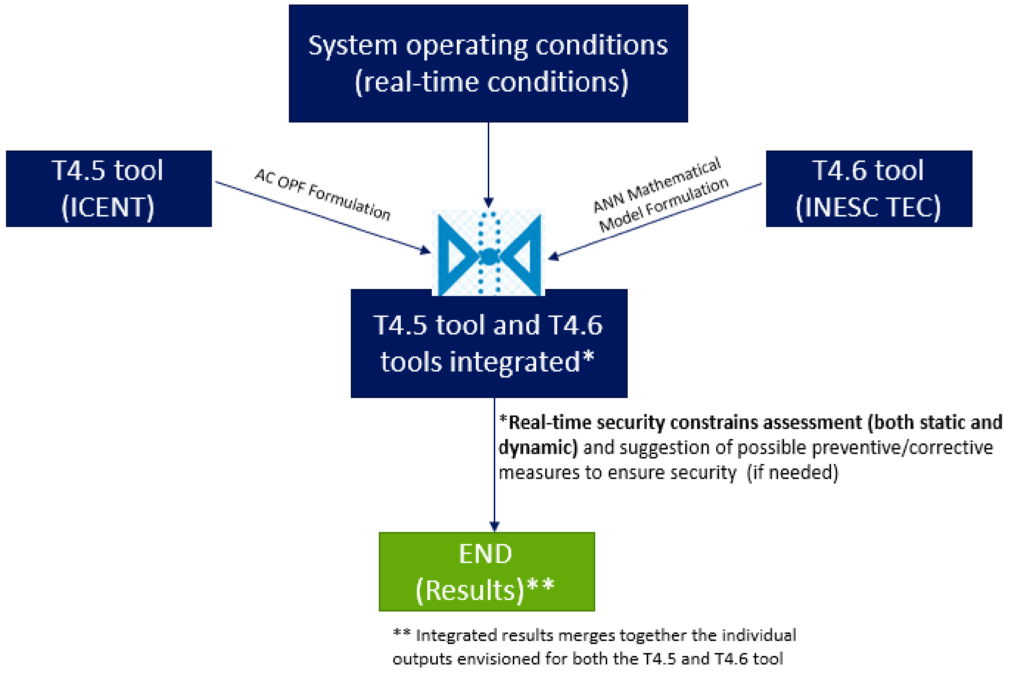


FIGURE 22: INTERACTIONS OF THE TOOL FOR DSA T4.6 WITH THE T4.5 TOOL CONSIDERING AN INTEGRATED VERSION OF BOTH TOOLS

5. References

[1] EU-ATTEST, "Deliverable D4.1 – Specification of the operation tools", June 2021. Available on: link.

[2] S. Liu et al., "An Integrated Scheme for Online Dynamic Security Assessment Based on Partial Mutual Information and Iterated Random Forest," in IEEE Transactions on Smart Grid, vol. 11, no. 4, pp. 3606-3619, July 2020, doi: 10.1109/TSG.2020.2991335.

[3] P. Demetriou, M. Asprou, J. Quiros-Tortos and E. Kyriakides, "Dynamic IEEE Test Systems for Transient Analysis," in IEEE Systems Journal, vol. 11, no. 4, pp. 2108-2117, Dec. 2017, doi: 10.1109/JSYST.2015.2444893.

[4] M. A. Pai: "Energy Function Analysis for Power System Stability", Kluwer Academic Publishers, Boston/Dordrecht/London, 1989, ISBN:0-7923-9035-0, ISBN-13: 978-1-4612-8903-6.

[5] T. Athay, R. Podmore, and S. Virmani, "A Practical Method for the Direct Analysis of Transient Stability," *in IEEE Transactions on Power Apparatus and Systems*, vol. PAS-98, no. 2, pp. 573-584., 1979.

[6] Distributed Electrical Systems Laboratory École Polytechnique Fédérale de Lausanne, "Extension of the IEEE 39-bus Test Network for the Study of Fundamental Dynamics of Modern Power Systems", Technical Report, February 2020.

[7] ENTSO-E, "Synchronous Condenser". Available on: link.

[8] H. Gu, R. Yan and T. K. Saha, "Minimum Synchronous Inertia Requirement of Renewable Power Systems," in IEEE Transactions on Power Systems, vol. 33, no. 2, pp. 1533-1543, March 2018, doi: 10.1109/TPWRS.2017.2720621.

[9] H. T. Nguyen, G. Yang, A. H. Nielsen and P. H. Jensen, "Combination of Synchronous Condenser and Synthetic Inertia for Frequency Stability Enhancement in Low-Inertia Systems," in IEEE Transactions on Sustainable Energy, vol. 10, no. 3, pp. 997-1005, July 2019, doi: 10.1109/TSTE.2018.2856938.

[10] H. Gu, R. Yan, T. K. Saha and E. Muljadi, "System Strength and Inertia Constrained Optimal Generator Dispatch Under High Renewable Penetration," in IEEE Transactions on Sustainable Energy, vol. 11, no. 4, pp. 2392-2406, Oct. 2020, doi: 10.1109/TSTE.2019.2957568.

[11] PSS/E Documentation, "Model Library PSS®E 34.5.1", 2018

[12] P. Kundur, N. J. Balu, and M. G. Lauby, Power system stability and control. McGraw-hill New York, 1994, vol. 7.

[13] "Determination of synchronous machine stability study models. volume 1", Westinghouse Electric Corp., Pittsburgh, PA (USA). Advanced Systems, Tech. Rep., 1981.

[14] "Determination of synchronous machine stability study models. volume 2", Westinghouse Electric Corp., Pittsburgh, PA (USA). Advanced Systems, Tech. Rep., 1981.

[15] "IEEE guide for synchronous generator modeling practices and applications in power system stability analyses", IEEE Std 1110-2002 (Revision of IEEE Std 1110-1991), pp. 1–72, 2003.

[16] IEEE draft recommended practice for excitation system models for power system stability studies", IEEE P421.5/D38, October 2015, pp. 1–202, Jan. 2015.

[17] Comission Regulation (EU) 2017/1485, "Establishing a guideline on electricity transmission system operation, 2 August 2017, Available on: link.

[18] ENTSO-E guidance document for national implementation for network codes on grid connection,

"Rate of Change of Frequency (RoCoF) withstand capability", January 2018, Available on: link.

[19] Wright, J.G. et al., "Formal comments on Integrated Resource Plan (IRP) 2018", 25 October 2018.

[20] ENTSO-E Report, "Future system inertia", Available on: link.

[21] Vasconcelos, H.; N. Fidalgo, J.; A. Peças Lopes, J., "A general approach for security monitoring and preventive control of networks with large wind power production" in Proc. PSCC 2002 – the 14th Power Systems Computation Conference, Seville, Spain, June 2002.

[22] A. Peças Lopes, J; Hatziargyriou, H.; Vasconcelos, N. Fidalgo, J.; Damianos, G.; Karapidakis, E., "Dynamic security evaluation functions in the more care project" in Proc. PSCC 2002 – the 14th Power Systems Computation Conference Proceedings of MedPower2002 - 3rd Mediterranean Conference and Exhibition on Power Generation, Transmission, Distribution and Energy Conversion, Athens, Greece, 2002.

[23] Lopes, J. A. Peças, N. Hatziargyriou, M. Vasconcelos, E. Karapidakis, and J. Fidalgo. "On-Line Dynamic Security Assessment of Isolated Networks Integrating Large Wind Power Production." Wind Engineering 23, no. 2 (1999): 107–17.

[24] H. Vasconcelos and J. A. P. Lopes, "ANN Design for Fast Security Evaluation of Interconnected Systems with Large Wind Power Production," *2006 International Conference on Probabilistic Methods Applied to Power Systems*, 2006, pp. 1-6, doi: 10.1109/PMAPS.2006.360206.

[25] Becker, P., "Recognition of Patterns", Polyteknish Forlag, Copenhagen, 1968.

[26] J. A. Pecas Lopes, F. P. Maciel Barbosa and J. P. Marques da Sa, "A two level hierarchical classifier for online transient stability assessment," *Proceedings. Electrotechnical Conference Integrating Research, Industry and Education in Energy and Communication Engineering'*, 1989, pp. 150-153, doi: 10.1109/MELCON.1989.50004.

[27] J. A. Peças Lopes, F. P. Maciel Barbosa & J. P. Marques de Sa, "On-Line Transient Stability Assessment and Enhancement by Pattern Recognition Techniques", in *Electric Machines & Power Systems*, vol. 15, pp. 4-5, 293-310, Jan. 2007, doi: 10.1080/07313568808909340

[28] M. Arai, ''Bounds on the number of hidden units in binary-valued three-layer neural networks,'' Neural Netw., vol. 6, no. 6, pp. 855–860, Jan. 1993.

[29] S. Tamura and M. Tateishi, ''Capabilities of a four-layered feedforward neural network: Four layers versus three,'' IEEE Trans. Neural Netw., vol. 8, no. 2, pp. 251–255, Mar. 1997, doi: 10.1109/72.557662.

[30] M. M. Islam and K. Murase, ''A new algorithm to design compact two-hidden-layer artificial neural networks,'' Neural Netw., vol. 14, no. 9, pp. 1265–1278, Nov. 2001, doi: 10.1016/S0893-6080(01)00075-2.

[31] M. I. C. Rachmatullah, J. Santoso and K. Surendro, "A Novel Approach in Determining Neural Networks Architecture to Classify Data With Large Number of Attributes," in *IEEE Access*, vol. 8, pp. 204728-204743, 2020, doi: 10.1109/ACCESS.2020.3036853.

[32] D. P. Kingma and J. L. Ba, "Adam: A Method for Stochastic Optimization," International Conference on Learning Representations (ICLR), 2015.