Electrical Power and Energy Systems 147 (2023) 108807

Contents lists available at ScienceDirect

International Journal of Electrical Power and Energy Systems

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

A fast data-driven topology identification method for dynamic state estimation applications

Davide Gotti *, Pablo Ledesma, Hortensia Amaris

Universidad Carlos III de Madrid, Av. Universidad 30, Leganés, 28911, Madrid, Spain

ARTICLE INFO

Keywords:
Topology identification
Dynamic state estimation
Deep neural network
Phasor measurement unit (PMU)
Bad data detection and identification

ABSTRACT

This paper proposes a fast topology identification method to avoid estimation errors caused by network topology changes. The algorithm applies a deep neural network to determine the switching state of the branches that are relevant for the execution of a dynamic state estimator. The proposed technique only requires data from the phasor measurement units (PMUs) that are used by the dynamic state estimator. The proposed methodology is demonstrated working in conjunction with a frequency divider-based synchronous machine rotor speed estimator. A centralized and a decentralized approach are proposed using a modified version of the New England test system and the Institute of Electrical and Electronics Engineers (IEEE) 118-bus test system, respectively. The numerical results in both test systems show that the method demonstrate the reliability and the low computational burden of the proposed algorithm. The method achieves a satisfactory speed, the decentralized approach simplifies the training process and the algorithm proves to be robust in the face of wrong input data.

1. Introduction

Dynamic state estimation (DSE) plays an increasingly important role in power system monitoring, control and protection [1,2]. The deployment of phasor measurement units (PMUs) in transmission networks [3,4] has enabled the development of fast DSE methods that improve situational awareness, opening new possibilities for power system protection techniques [5]. Particularly, the estimation of synchronous generator rotor angles and speeds can be used to implement wide-area protection schemes based on out-of-step protections [6,7] or on direct transient stability methods such as the energy function. The estimation of rotor speeds can be used also to minimize the amount of load or generation to be shed after a load imbalance to preserve the integrity of the system [1].

DSE algorithms such as [8–12] need reliable topology information to be properly executed. The topology configuration is provided by the network topology processor, which performs the topology identification (TI) process. Traditional TI relies on the status of the electrical switching devices and can be affected by malfunctions, as described in [13]. TI algorithms for fast DSE face the additional problem of having to be executed in a short time, typically a few tenths of milliseconds if the result is to be used in a protection scheme.

Several TI methods have been developed in recent years, most of them implemented in the framework of power system state estimation in energy management systems. This is the case of the residual analysis method and the state vector augmentation method, both described in [13]. These methods require one or several state estimation steps to execute topology processing, which is a limiting factor for online applications. The algorithm described in [14] proposes the usage of PMUs to estimate the admittance matrix of the network. Although reported results are accurate, this method requires to monitor each network bus with a PMU or a modern relay. Currently, not many electrical grids are equipped with such a great number of monitoring devices, since its implementation is still rather costly. A methodology that includes the status of the switching devices in the state estimation vector is proposed in [15]. This implies the inclusion of three new variables in the state vector for each switching device, increasing the computational time of the algorithm. In [16], normalized Lagrange multipliers are used to detect topology errors along with bad data. The method presents accurate results, but the reported computational time is well above one second, making the algorithm inappropriate for fast DSE applications.

The usage of neural networks in TI applications has been introduced in pioneering work presented in [17,18]. In [17], a counter propagation neural network that makes use of the power flow and injection measurements, along with the switching status of several branches, is proposed. In case the switching status provided by the network topology processor is affected by errors, which is not a remote possibility [13], the accuracy of this methodology could be greatly affected.

* Corresponding author.
E-mail addresses: dgotti@ing.uc3m.es (D. Gotti), pablole@ing.uc3m.es (P. Ledesma), hamaris@ing.uc3m.es (H. Amaris).

https://doi.org/10.1016/j.ijepes.2022.108807
Received 25 January 2022; Received in revised form 9 October 2022; Accepted 18 November 2022
Available online 6 December 2022

The usage of artificial neural networks (ANN) for topology processing, along with the bad data detection and identification functionality, is proposed in [18]. The methodology requires the implementation of four different ANNs for each branch, and a previous state estimation forecasting step to distinguish between topological and gross measurement errors. In case large power systems are considered, this methodology could lead to a significant number of ANNs to be trained and tested. On the contrary, with our approach, the topology identification can be carried out using only one DNN, leading to a more straightforward approach. Furthermore, no state pre-estimation step is needed, which translates into a significant reduction of the computational time. A robust principal component analysis coupled deep belief network for topology identification has been recently proposed in [19]. Reported results show high accuracy, good robustness to measurement noise and missing data, and computational times are of the order of tens of ms. However, this methodology relies on the relations between bus voltages to identify the branch status. While in distribution networks with a typical radial connection between nodes these features can be a reliable indicator of the branch connections, in meshed transmission networks this approach might not be reliable. In [20], a hierarchical framework that makes use of recurrent neural networks to identify line outages is proposed. Reported results are very accurate, but the reported computational times are of the order of hundreds of ms, which can be a limiting factor for DSE applications. In [21], the TI is provided by a neural network taking measurements of voltages, power injections and power flows from a wide area. However, since this method receives the measurements from both traditional instrumentation and PMUs with different sampling times, it is not applicable with fast DSE applications. Furthermore, its training process can become cumbersome if applied to large electric networks. A state and topology estimation for distribution systems is proposed in [22]. Different DNNs are proposed to carry out the state estimation and the TI separately. A sequential forward selection is used to conduct the PMU placement. Subsequently, their measurements are used to train the DNNs and to conduct the estimates during the real-time operation. The method shows very high TI accuracy and it can be applied to unbalanced systems. However, the methodology has not been tested in transmission systems under strong perturbations such as three-phase faults and under the presence of bad data in the measurement set.

Apart from neural network-based algorithms, several other data-driven techniques have been recently proposed to solve the topology recovery problem. The methodology proposed in [23] carries out the TI by employing a fuzzy c-means clustering method. This method is reliable and accurate if applied to relatively small networks, but suffers from loss of accuracy when the size of the grid and the number of topology configurations to be estimated increase. Furthermore, this method needs the input measurement set used to conduct the state estimation of network bus voltage phasors, and therefore it is not well suited for DSE applications. An event-triggered TI based on a recursive Bayesian approach is proposed in [24]. This method is sensitive to high measurement noise levels and the topology processing is reliable, although the number of possible topology configurations is limited. In [25], a two-step method to carry out topology identification is introduced. Firstly, a data-driven approach is used to carry out a first approximation of the system admittance matrix. Secondly, a model-driven formulation is employed to improve the estimation of the admittance matrix. Although reported results show high accuracy, the computational times range from few seconds up to 10 s. Whereas this computational speed might be acceptable for the monitoring and control of distribution systems, it is not admissible for DSE applications. A split expectation–maximization data-driven approach for topology identification has been recently introduced in [26]. In the first stage, an historical data batch topology identification is carried out to identify the number of possible topology categories. Subsequently, several classifiers based on machine learning methods are proposed for real-time implementation. Reported results are accurate, but simulations are conducted only under normal operating conditions. Therefore, it is not possible to determine whether the method is reliable or not under dynamic conditions. Furthermore, reported results exhibit CPU times on the order of tens of milliseconds, whereas the method proposed in this work is significantly faster. In [27], a two-stage topology identification algorithm is proposed. In the first stage, a mixed-integer programming model is used to carry out a preliminary TI. In the second stage, a spanning tree generation algorithm is used to search for the topology configuration that best matches the obtained estimations with power-flow results. Reported computational times are ranged between some seconds to some minutes, and this clearly poses a limitation for DSE applications.

Table 1
Time requirements of DSE for different applications [5].

DSE application	Time requirements (s)	Measurement
Transient stability	$\approx 10^{-2}$	PMU
Frequency stability	$\approx 10^{-1}$	PMU
Long-term voltage stability	$\approx 10^{-1}$	PMU
Short-term voltage stability	$\approx 10^{-2}$	PMU
Control of converter-based resources	$\approx 10^{-4}$	SV
Out-of-step protection	10^{-2}–10^{-1}	PMU
Fault location	$\approx 10^{-3}$	SV

Generally, the aforementioned methods show computational times ranging between hundreds of milliseconds and some seconds. Table 1 summarizes the time requirements for different DSE applications and the technology used to collect the input measurement, e.g., PMU or synchronized sampled value (SV) measurements. It can be seen that most of the TI methods previously described do not meet the real-time requirements for DSE applications. The interested reader can refer to [5] and the references within for further details.

In order to obtain the necessary speed and robustness, TI methods must be adapted to the DSEs they are intended to serve. Many DSEs based on different Kalman filter types have been recently proposed, e.g., [28–32]. All these methods require, at least to some extent, knowledge of the synchronous machine parameters, which can be a limiting factor. Recent years have seen the introduction of the concept of the frequency divider (FD) formula [33], which allows calculating the frequency of the center of inertia [8], the synchronous generator rotor speeds [9], the rate of change of power (RoCoP) [10,11], and to carry out an online inertia estimation of synchronous and non-synchronous devices [12]. FD-based methods make use of the frequency measurements provided by PMUs and are characterized by being model-independent. In fact, modeling of generators and loads is not required because their interactions with the rest of the system are reflected in the bus frequency variations captured by the PMUs. These algorithms are characterized by having a very small computational burden [34]. However, incorrect topology identification can heavily affect their estimations because these algorithms rely on several local measurements and the correct status of some relevant branches, as explained in Section 2.4.

This work proposes a fast TI method to avoid estimation errors under changes in the network topology. The main challenge is achieving a reliable estimation of the topology sufficiently fast to be used together with online state estimators. An additional goal is improving the speed and robustness of the estimator by avoiding communications with a central control room. The proposed method is based on a deep neural network (DNN) that takes voltage and current measurements provided by PMUs as input data.

The main advantages of the proposed method compared to other methods in the literature are:

- It can be coupled with fast dynamic state estimators in online applications since its computational burden is very small. To the best of the authors' knowledge, no other TI method presents lower CPU times.

- It provides accurate TI even under the occurrence of strong perturbations in the electric system.
- It does not require the installation of any additional measurement devices, since it uses the same PMU measurements needed to conduct the DSE. In this regard, it is worth remarking that the presence of a PMU at the point of connection of a power plant is recommended by NERC standards for better power system operation and control [35] and device model validation [36].
- It is suitable for large power networks since it can be applied following a decentralized approach, avoiding the cumbersome training process that limits the applicability of a centralized implementation to large electric networks.
- It can be coupled with fast bad data detection, identification, and substitution methods that allow the proposed TI to be robust to gross measurement errors.

The proposed TI method is demonstrated together with a bad data detection, identification, and substitution (BDDIS) algorithm and a FD-based rotor speed estimator, both of which use the same input data and similarly short execution times. The simulations are run on a modified version of the New England test system and on a portion of the The Institute of Electrical and Electronics Engineers (IEEE) 118-Bus system. However, the output of the proposed TI method can be used to feed other DSEs that rely on the correct network topology.

The rest of the paper is organized as follows. Section 2 describes the algorithms used in this work. Sections 3 and 4 demonstrate the proposed method using a centralized and a decentralized approach, respectively. Finally, Section 5 concludes the paper.

2. Method description

The proposed method uses a DNN to execute the topology identification, solving a classification problem. The final aim of this formulation is to avoid a biased error of the model-independent linear generator rotor speed estimator. The general methodology is illustrated in Fig. 1. By operating on voltage and current data received from PMUs, which can be refreshed every few milliseconds, it is possible to take full advantage of the speed of the proposed method. A separate BDDIS module is responsible for the detection of bad data, liberating the TI from this task and simplifying the training process of the DNN.

The proposed approach is based on DNNs because of their high accuracy and small computational cost. Other machine and deep learning methods such as the Naïve Bayes method, the decision trees, and linear and logistic regression algorithms have comparable computational times, but they are outperformed by DNNs in terms of estimation accuracy [37]. On the other hand, algorithms such as decision forests and support vector machines provide similar estimation accuracy to DNNs, but are significantly slower [37] and therefore cannot be used for fast DSE applications.

2.1. Deep neural networks

DNNs are characterized by having several hidden layers between the input and output neurons. Generally, they can accomplish more complex tasks as their dimension increases.

Before the DNN can be used to execute a prediction, it must be properly trained. During the training process, the input signal is propagated forward through the synaptic connections of the DNN until an output signal is generated at the output end of the structure. Then the output signal is compared with a desired output, and an error signal is calculated based on the error signal function. This error signal is back-propagated through the DNN to adjust the synaptic weights and the neuron bias and, consequently, to minimize the cost function of the error signal. The iterative process is repeated until an acceptable accuracy on the training set is obtained. The conceptual steps of the training process are briefly reported hereafter, but the interested reader can refer to [38,39] for a comprehensive explanation of DNNs and their optimization heuristics.

Feed-forward propagation

In the feed-forward step, the input signal is propagated from one neuron to the next layer neurons through the synaptic connections:

$$\boldsymbol{x}_i = \boldsymbol{W}_{i,i-1}\boldsymbol{y}_{i-1} + \boldsymbol{b}_i, \tag{1}$$

where $\boldsymbol{x}_i$ represents the input vector entering the i_{th} layer, the matrix of (n, m) dimensions $\boldsymbol{W}_{i,i-1}$ represents the synaptic weights (where n and m are the number of neurons of the i_{th} and $i-1_{th}$ layers, respectively), vector $\boldsymbol{y}_{i-1}$ represents the output signal coming out of the $i-1_{th}$ layer, and vector $\boldsymbol{b}_i$ represents the neuron bias values to be applied to the neurons in the i_{th} layer.

Afterwards, the computation of the active function is performed, which is usually a nonlinear function of the input pattern:

$$\boldsymbol{y}_i = \boldsymbol{a}_i(\boldsymbol{x}_i), \tag{2}$$

with $\boldsymbol{a}_i$ being the neuron activation function, which generates the i_{th} layer output vector $\boldsymbol{y}_i$.

The activation function strongly modifies how the output signal is computed and the accuracy of the DNN on both the training and test phases; therefore, its formulation must be carefully evaluated [39]. Finally, when the input vector is propagated through all the hidden and output layers, the error signal can be calculated as follows:

$$\boldsymbol{e}_j = \boldsymbol{\zeta}(\boldsymbol{o}_j, \hat{\boldsymbol{o}}_j), \tag{3}$$

where $\boldsymbol{e}_j$ is the error vector, j represents the output layer index, and $\boldsymbol{\zeta}$ is a generic function that computes the output error based on the difference between the estimated output vector $\boldsymbol{o}$ and the desired output vector $\hat{\boldsymbol{o}}$.

As in the case of the activation functions, the error function formulations also have an important impact on the DNN performance. The interested reader is directed to [39] for a detailed discussion on this topic.

Backpropagation

During the backpropagation step, the partial derivative of the error signal with respect to the synaptic weights and bias are determined. In this way, it is possible to compute the adjustments in the weights and bias hyperspace that minimize the error signal. The general formulation, applied to the hidden layers, is the following:

$$\frac{\partial \boldsymbol{\xi}_i}{\partial \boldsymbol{W}_{i,i-1}} = \frac{\partial \boldsymbol{\xi}_i}{\partial \boldsymbol{e}_i}\frac{\partial \boldsymbol{e}_i}{\partial \boldsymbol{y}_i}\frac{\partial \boldsymbol{y}_i}{\partial \boldsymbol{x}_i}\frac{\partial \boldsymbol{x}_i}{\partial \boldsymbol{W}_{i,i-1}}, \tag{4}$$

where ξ is the error signal backpropagated up to the i_{th} hidden layer. In the case of the output layer, since there is a direct relationship between the error signal and the neurons, the formulation is simpler:

$$\frac{\partial \boldsymbol{e}_j}{\partial \boldsymbol{W}_{j,j-1}} = \frac{\partial \boldsymbol{e}_j}{\partial \boldsymbol{o}_j}\frac{\partial \boldsymbol{o}_j}{\partial \boldsymbol{x}_j}\frac{\partial \boldsymbol{x}_j}{\partial \boldsymbol{W}_{j,j-1}}, \tag{5}$$

with j being the output layer index. Once the partial derivatives are computed, the weights can be updated:

$$\boldsymbol{W}_{i,i-1} = \boldsymbol{W}_{i,i-1} - l_r\frac{\partial \boldsymbol{\xi}_i}{\partial \boldsymbol{W}_{i,i-1}}, \tag{6}$$

where l_r is the learning rate, which allows us to tune the trajectory rate of change in the weight and bias hyperspace. To update the neuron bias, an analogous procedure is executed.

Once the DNN achieves the desired accuracy over the training set, its performance must be tested. If the training process has been carried out preventing overfitting and underfitting phenomena, the DNN is able to correctly execute the predictions outside the training set.

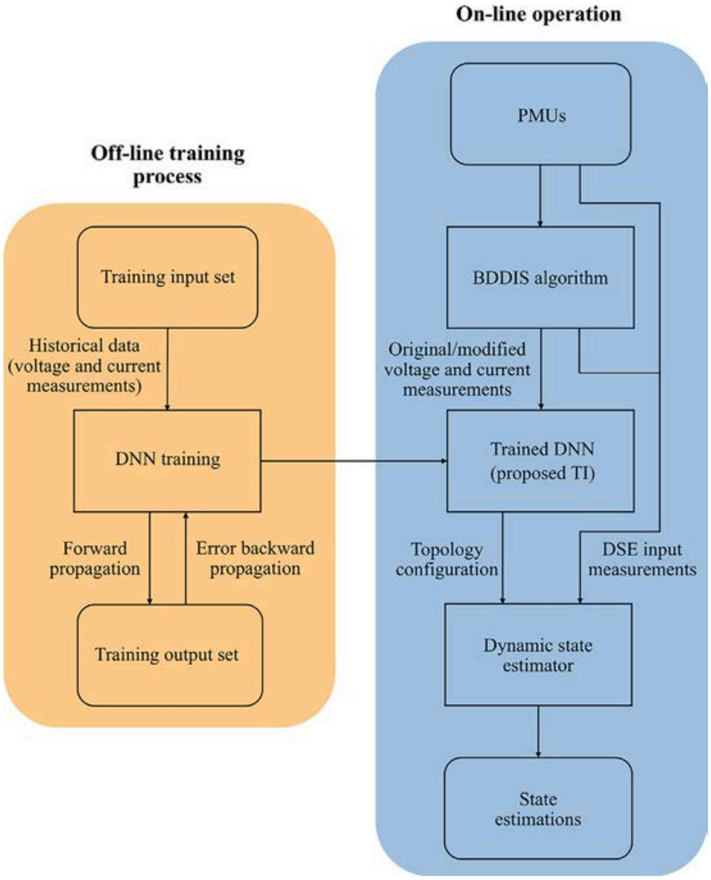


Fig. 1. Flowchart of the proposed methodology.

2.2. Proposed TI method

The input vector is composed of the bus voltages and the branch currents provided by the same PMUs used to carry out the dynamic state estimation. Before the measurements are used by the TI, they are normalized as follows:

$$x_{i_{normalized}} = \frac{x_i - x_{min}}{x_{max} - x_{min}}, \tag{7}$$

where x_i is the i_{th} measurement of the input vector, and x_{max} and x_{min} are the maximum and the minimum values of the training set, respectively, to be applied separately for each type of measurement. As explained in [21], this procedure avoids generalization problems and typically accelerates the training process. Once the input signal has propagated through the DNN, the output signal is obtained at the output end of the structure. Depending on the number of possible topology configurations, a different number of output neurons must be used. For example, if the number of topology configurations is between 9 and 16, then a 4-digit binary number must be used. The topology configurations considered for the centralized and decentralized case studies in the following sections can be found in [40]. In this work, all the hidden and output layer neurons employ the sigmoid activation function:

$$y_i = \frac{1}{1 + e^{-x_i}}. \tag{8}$$

Several other activation functions are commonly used in DNN applications, e.g., the rectified linear activation (ReLU) function, the hyperbolic tangent function, and the softmax function. As described in Section 2.1, the choice of the activation function plays an important role in the performance of DNNs, and it must be selected following deep learning heuristics and trial and error procedures [39]. In this work the ReLU and the hyperbolic tangent functions were also tested on hidden layer neurons. However, the ReLU activation function was ruled out because it experienced the so-called “dying ReLU problem”. This phenomenon occurs when a significant number of internal signals in a DNN are negative and the neuron outputs are null, thus preventing the DNN from learning. The hyperbolic tangent function was also discarded because it provided a lower accuracy in the test phase compared to the sigmoid function. Finally, the sigmoid activation function is selected for output layer neurons because this function is equivalent to a 2-class softmax function [39] and therefore best suited for the binary classification problem formulated to identify the electric grid topology.

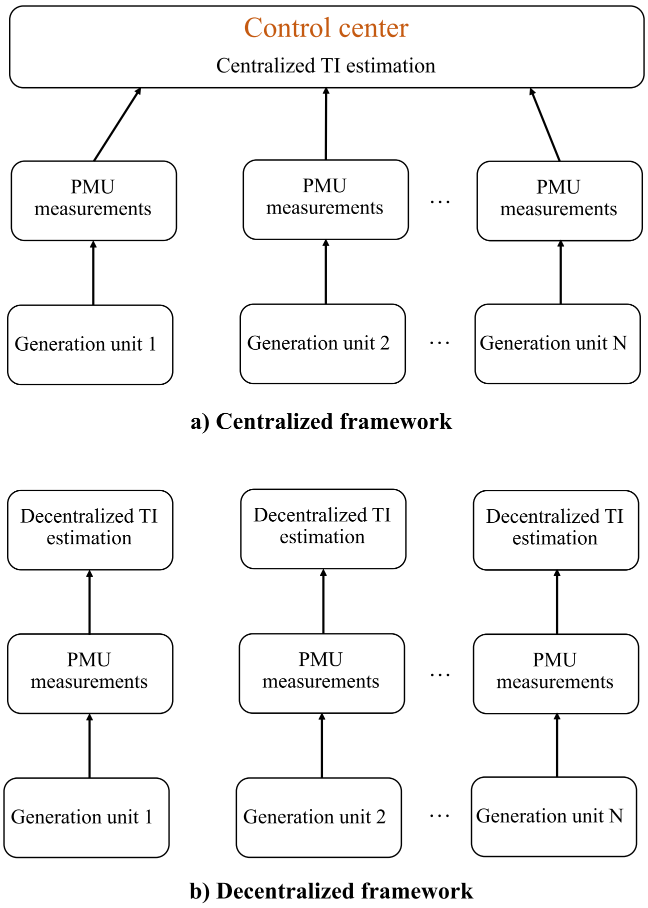
The formulation of the error signal used to train the DNN is the following:

$$e_j = (o_j - \hat{o}_j)^2. \tag{9}$$

The calculation of the admittance matrix requires several seconds and has to be done only once for each topology configuration, unless there are major changes such as the substitution of a transformer or a line. Consequently, the admittance matrix corresponding to each topology configuration is calculated offline.

The proposed algorithm is applicable using a centralized or a decentralized implementation. Since it can significantly affect the practical application of the proposed method, its main advantages and disadvantages are described hereunder and illustrated in Fig. 2.

Centralized TI approach. The centralized TI approach allows the algorithm to determine the relevant topology configurations for the generator rotor speed estimations using only one DNN. It is a straightforward implementation, but it may not be applicable in large power systems since the number of possible topology configurations grows exponentially. This fact significantly increases the training times and the possibility that the DNN experiences overfitting or underfitting



a) Centralized framework

b) Decentralized framework

Fig. 2. Centralized and decentralized implementation of the proposed TI method.

phenomena. In the experience of the authors of this work, the centralized approach is advisable up to a few tens of possible topology configurations. After that, a decentralized implementation is preferable.

Decentralized TI approach. For large power systems, the training process of the DNN can be time consuming since the number of relevant topology configurations increases drastically. As previously described, this fact can lead to a cumbersome training process. Fortunately, since the estimation algorithm [9] allows a decentralized estimation of the rotor speed, topology processing can also be employed in a decentralized way, i.e., it can be applied individually to each synchronous generator. Since only the branches that connect the generation bus with its neighboring buses must be taken into account, this implementation minimizes the number of possible topology configurations. Thus, the training set is smaller, and the training process is significantly easier. In fact, as described in [39], the training process of classification DNNs depends, among other parameters, on the size of the training set and the number of possible outputs of the DNN. By dividing the topology processing into several minor TI tasks, we reduce both the training set of each DNN and the number of possible topology configurations to be estimated. As a result the training time is remarkably reduced, as shown later in Section 4. For its practical relevance, it is worth mentioning that also other FD-based algorithms allow decoupled estimations and can therefore benefit from the decentralized implementation of the proposed TI. For instance, both RoCoP [10,11] and inertia [12] estimations could be coupled with the proposed decentralized approach.

2.3. Bad data detection, identification, and substitution algorithm

In case the input signal of the proposed TI method is affected by gross errors, the topology processing might be biased. Since DNNs need a fixed number of input values, it is not possible to simply identify and remove the measurement. Instead, a BDDIS algorithm should be used. In this article, due to its low computational cost, a modified version of the approach proposed in [41] is applied. In this way, the BDDIS is carried out using a neural network pre-estimation filter. With this methodology, the BDDIS receives the input measurement set and provides their estimated values. The output and input training sets are exactly the same, and both sets consist of correct measurements of the considered system operating condition. During the DNN training process, the features between input and output signals are extrapolated by computing the synaptic weight and neuron bias values. In this manner, the presence of bad data in the input set is identified and the DNN provides the closest match to these inputs to generate an educated guess.

To identify the bad data, the following identification rule is used:

$$(z_i - \phi_i)^2 > {\gamma_i}^2, \tag{10}$$

where z_i and ϕ_i are the i_{th} measurement and its estimated value provided by the DNN, respectively, and γ_i is the threshold that flags the presence of bad data in the measurement set.

In case the i_{th} measurement is flagged, the input value z_i is replaced with its estimation ϕ_i provided by the DNN. As the estimated value is generally similar to the true input value, it can be used to conduct the TI.

It is worth mentioning that in order to conduct the BDDIS process, other algorithms can be used. An extensive survey of possible data-driven algorithms that can be used in power system applications is presented in [42]. For instance, a convolutional neural network is presented in [43], which shows high detection accuracy against false data injection attacks. A nonlinear autoregressive exogenous neural network is presented in [44], which presents small running times and high bad data detection accuracy. Other algorithms that can be used for the BDDIS process are [45,46], since they are both highly accurate and present CPU times that are compatible with the time requirements of fast DSE algorithms.

2.4. FD-based rotor speed estimator

In order to demonstrate the proposed TI method, it is coupled with a rotor speed estimator based on the model-agnostic linear algorithm introduced in [9]. The rotor speed estimator is based on the frequency variations measured by the PMUs, and applies a weighted least square (WLS) formulation that results in the following equation:

$$\Delta\omega_G = (\boldsymbol{D}^T\boldsymbol{D})^{-1}\boldsymbol{D}^T\Delta\omega_B = \boldsymbol{D}^+\Delta\omega_B, \tag{11}$$

where $\boldsymbol{D}$ is the frequency divider matrix [33] and $\boldsymbol{D}^+$ is its left inverse, vector $\Delta\omega_G$ represents the sought generator rotor speed deviations, and vector $\Delta\omega_B$ represents the bus frequency deviations measured by the PMUs.

The WLS formulation allows a fully decoupled estimation of the rotor speed of each generator. In fact, matrix $\boldsymbol{D}^+$ presents interesting topology properties, since the non-null elements of each row of $\boldsymbol{D}^+$ correspond to the bus frequency measurements required to estimate the generator rotor speed associated with the analyzed row. More specifically the relevant frequency measurements refer to the bus of the generator to be monitored and its neighboring buses. This is an important consideration, because by using this property the topology processing can also be applied in a decentralized way, i.e., considering only the branches that are *relevant* for the estimation of the rotor speed of each generator. This consideration is valid also for other FD-based estimation algorithms, such as [8,10–12].

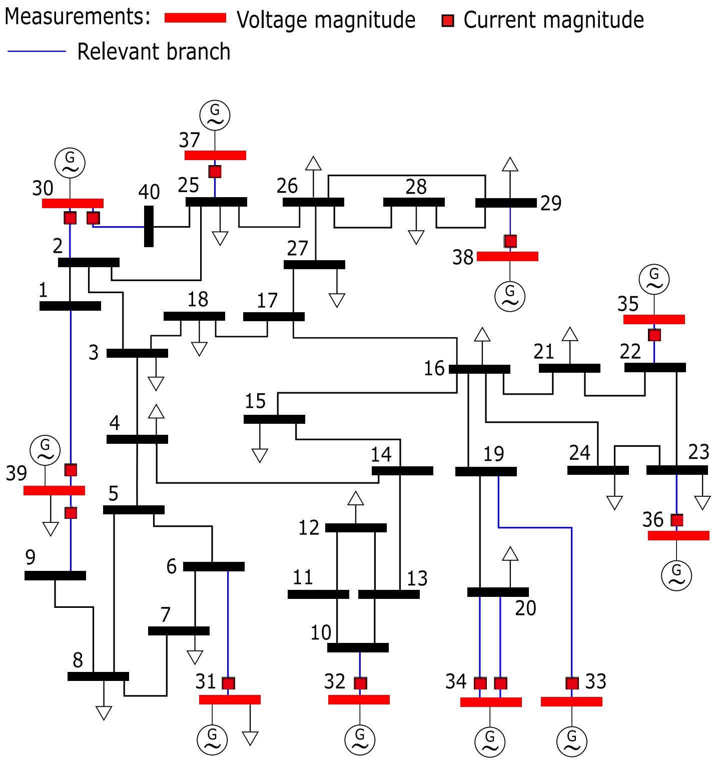


Fig. 3. Single-line diagram with the PMUs used for TI and the relevant branches of the modified New England test system.

3. Study case 1: Centralized application on the New England test system

3.1. Description

The proposed TI method is firstly demonstrated on a modified version of the New England test system, shown in Fig. 3. This is a well known benchmark system that has been selected because it is sufficiently small to demonstrate the centralize application of the proposed TI method. The computational load of the training process increases rapidly with the size of the network; in the case of large interconnected networks it is recommended to follow the distributed approach described in Section 4.

The relevant branches whose status needs to be estimated to correctly execute the FD-based algorithm are painted in blue. All synchronous generator connection points are equipped with PMUs that measure the bus voltage and the branch current; altogether, a set of 23 PMU measurements is used to feed the DNN. A detailed account of the power system model and the set of measurements can be found in [40].

Generators in the New England test system are connected in antenna. In these cases, according to the properties of the WLS estimator we only need the current through the branch connecting the generator to the rest of the system to estimate the rotor speed. If the branch opens the generator remains isolated from the grid and it simply trips. More interesting cases occur when the generator is connected with two transformers working in parallel and when the generator is connected to two different buses. To represent these last situations, the New England test bus system has been sightly modified: the branch connecting buses 20 and 34 has been split in two parallel transformers, and a new bus 40 has been added. This new bus 40 is connected to bus 30 through a transformer and to bus 25 through a line. Since the relevant branches for the considered DSE are the ones connecting the generation buses with their neighboring nodes, one may come up with the idea to estimate the connection status by relying solely on the PMU current measurements. This method would probably lead to incorrect TI estimations in case bad data was present in the measurement set since the TI is based on a single measurement value. Since DNNs are able to extract the features between the input values and do not rely on a single measurement value, they are intrinsically more robust to the presence of outliers in the data set. However, as mentioned in Section 2.3, the presence of a BDDIS remains recommended to further increase the robustness against outliers.

Table 2
Training and test accuracy of the DNN centralized approach.

	Centralized approach DNN with 3 hidden layers 23-30-20-10-4
Training accuracy	99.81%
Training time	1375 s
Testing accuracy	99.43%
Testing time	$5.1 \cdot 10^{-5}$ s

Because the test system is relatively small, a centralized approach can be implemented by applying a unique DNN that is able to recognize the relevant topology configurations. The DNN implemented for this case is able to detect significant topological changes for each generator. In total, it is trained to recognize 16 possible configurations, which are reported in [40]. Table 2 summarizes the parameters used to build, train and test the DNN, which have been determined heuristically. The DNN, in the centralized case, has 23 input neurons, 3 hidden layers with 30, 20 and 10 neurons, and 4 output neurons. The learning rate is $lr = 5 \cdot 10^{-5}$. It is worth mentioning that the structure of the DNN has been determined empirically after several attempts. In accordance with [38-39], we noticed that smaller structures with a smaller number of hidden layers and neurons could not reach a good training accuracy even after large training times, since they were not capable of extracting the relevant feature from the data set. On the contrary, bigger structures showed high training accuracy but experienced overfitting problems and performed poorly outside the training set during the test phase.

3.2. Training process and test results

The training process of the DNN is executed considering generation and load events. These events produce fluctuations in the electric network so that the bus voltage varies from 0.8 p.u. to 1.20 p.u., and the branch current flows from approximately 0 to 120%. These variations allow the DNN to be properly trained and to extract the relevant features that enable correct input–output mapping. The DNN of the centralized approach is trained considering 145 generation and load events, and its training set is composed of 5648 samples. In case the DNN does not perform acceptably well during the test phase with different DNN structures, it is necessary to modify the training data set by considering new load and generation events so that the DNN can extract the relevant features from the data and correctly carry out the TI.

The DNN testing phase is executed considering a topology change following a short-circuit clearance. All the simulations last 10 s, with the short-circuit occurring at t=1 s, and the fault being cleared 100 ms later. The simulation time step used is 10 ms, therefore the DNN executes 1000 topology processing for each simulation. This scenario is repeated considering every relevant topology configuration. The set of voltage and current measurements used for the DNN training and test phases are imported from the software PowerFactory [47].

To test the sensitivity of the proposed TI method to the presence of noise, a Gaussian distributed error with zero mean and σ standard deviation is added to all the measurements used to execute the topology processing. The accuracy of the measurement devices corresponds to 3σ, and the value for this work is taken from [48]: $3\sigma = 0.7\%$ of the reading value for both the bus voltage magnitude and line current magnitude measurements. It is worth mentioning that a recent study [49] proved that PMU measurements present a non-Gaussian distributed noise. However, the methodology presented in this work remains valid since DNNs are able to perform accurately with measurements affected by a generally distributed noise if that noise distribution is taken into account during the training phase [50]. This has already been proven also in [22], where a non-Gaussian noise is considered and the proposed DNN is able to correctly carry out the TI classification.

Table 2 shows the accuracy achieved during the test phase, which is higher than 99%. The only situations in which the proposed TI method does not recognize the topology occur throughout the short-circuit event. In fact, during the training process these events were not considered, and it is therefore predictable that the DNN cannot properly interpret them. As can be noticed, the topology processing time is in the order of 10^{-5} s per iteration.

Table 3 shows the most important performance indicators from the point of view of the final user in comparison with the TI methods described in Section 1. Comparing Tables 1 and 3, it can be seen that most of the TI algorithms available in literature are not suitable for DSE applications, since they do not meet the execution time requirements. Only Refs. [19,21,23], and [26] exhibit CPU times that are compatible with some execution times in Table 1; however, as described in Section 1, these methods are not suited for DSE applications. In this regard, the proposed TI method is promising because its low computational burden makes it compatible with fast DSE applications.

3.3. Simulation results

The proposed TI method is integrated with the WLS estimator described in Section 2 to demonstrate how the latter can benefit from a correct TI. The input frequency measurements and the actual rotor speeds (taken as reference output values) are imported from the software PowerFactory [47]. The model used by PowerFactory to compute the bus frequency values, used as input values of the simulations, is a classical phase-locked loop (PLL).

No noise for the bus frequency measurement is supposed, since the WLS rotor speed estimator used in this work is quite sensitive to these types of turbulence and the results may be significantly affected by this factor. The effect of the noise on the speed estimator can be decreased through appropriate filtering [9] and is beyond the scope of this paper.

Table 3
Performance comparison of different TI algorithms.

Method	Training time	Execution time (s)	Topology identification accuracy (%)
[16]	Not applicable	$\approx 10^{0}$	100
[17]	3–5 h	10^{-1}–10^{0}	100
[19]	30 min	$\approx 10^{-2}$	≈ 99.9
[20]	Not reported	10^{-1}–10^{0}	98–100
[21]	10–90 h	$\approx 10^{-4}$	99.6–99.9
[23]	Not applicable	$\approx 10^{-2}$	100
[24]	Not applicable	10^{0}–10^{1}	62.5–100
[25]	Not applicable	10^{0}–10^{1}	100
[26]	Few minutes	$\approx 10^{-2}$	≈ 99.4
[27]	Not applicable	$\approx 10^{1}$	90–100
This work	1–30 min	$\approx 10^{-5}$	≈ 99.4

To quantitatively compare the results obtained using the proposed TI method with the case in which no TI is considered, the root mean square (RMS) of the residuals is calculated as:

$$\Psi_{\omega_{Gi}} = \sqrt{\frac{1}{T}\sum_{t=1}^{T}[\omega_{\hat{Gi},t} - \omega_{Gi,t}]^2}, \tag{12}$$

where $\omega_{\hat{Gi}}$ is the estimation of the i_{th} generator rotor speed, t represents the considered time step, T is the number of time steps of the simulation, and ω_{Gi} is the actual generator rotor speed value.

In all the simulations reported in Sections 3 and 4, the time step is 10 ms, and random Gaussian noise, as described in Section 3.2, is applied to the measurements used to feed the DNN. The algorithms are coded in MATLAB and run on a computer with an Intel Core i7-3770, 3.40 GHz processor and 6 GB of RAM.

3.3.1. Short-circuit at bus 34 followed by disconnection of transformer 20-34

The simulated case consists of a three-phase short-circuit with zero fault impedance at t=1 s located at the low voltage side of one of the two parallel transformers between buses 34 and 20. The fault is cleared by the disconnection of the affected transformer 100 ms later. This event leads to a topology change that affects the elements (5,20) and (5,34) of D^{+} [40] and therefore affects the rotor speed estimation of the generator at bus 34.

In Fig. 4 the results of the topology processing at each iteration are reported. As previously discussed, the proposed TI method employs a binary classification, and the results are reported in this format [40]. As can be noticed, the proposed method provides accurate TI along the simulation, correctly detecting that the topology is not changing during the first part of the simulation (output “0000”) and detecting the opening of one of the two parallel transformers between buses 20 and 34 (output “1010”). Only during the fault, there are few misclassifications, since the algorithm provides the output “1000” (transformers 2-30 and 40-30 are both open). It is worth mentioning that the proposed TI method is very accurate under the severe electromechanical oscillations caused by a nearby three-phase short-circuit and its clearance.

Fig. 5 shows the actual rotor speed of the considered synchronous machine and the estimated rotor speed with and without the proposed TI. It can be seen that the proposed method provides more accurate estimates of the generator rotor speed due to prompt detection of the new topology configuration. It is worth mentioning that the spikes observed during the fault occurrence and its clearance are due to the frequency calculation technique of the PMUs. In this work, as already mentioned, they are provided by the software PowerFactory and are based on a PLL model [47].

Observing the last instants of the simulation, it can be noticed how both rotor speed estimations converge to the actual rotor speed values.

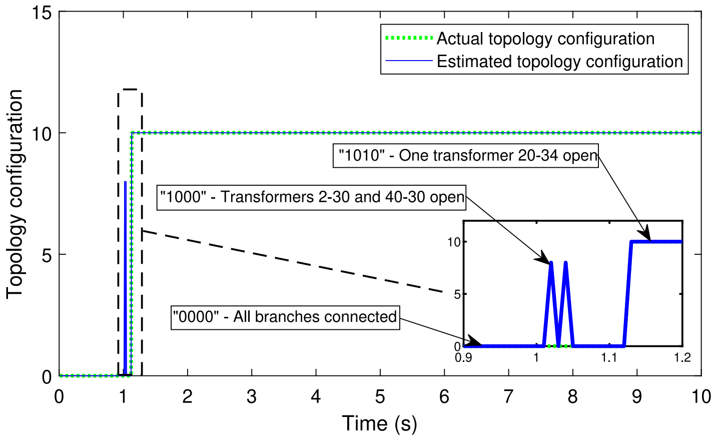


Fig. 4. Case 1-1 — Topology identification.

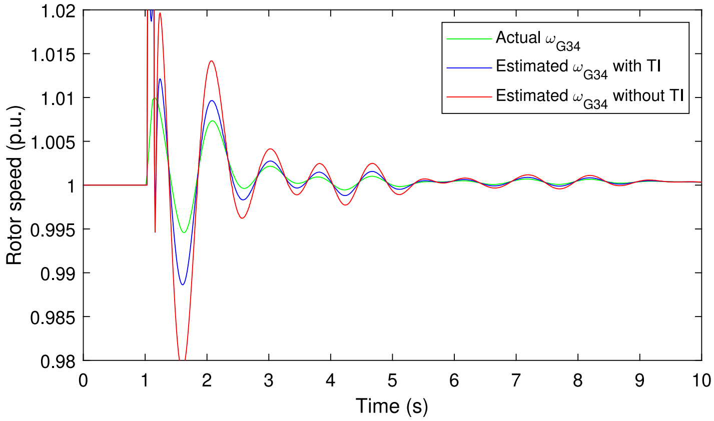


Fig. 5. Case 1-1 — Rotor speed estimation of synchronous machine 34.

Table 4
Case study 1-1.

Comparison index	Rotor speed estimation with TI	Rotor speed estimation without TI
$\Psi_{\omega_{G34}}$ (p.u.)	$2.106 \cdot 10^{-3}$	$4.321 \cdot 10^{-3}$
CPU time (s)	2.71	2.67

The benefit of using correct topology information manifests only during the first seconds after the fault. In fact, it is only during this time lapse that the bus frequencies vary, and the weight given by the non-null elements of D^{+} affects the estimation of the rotor speed estimations.

With the proposed TI method, the enhanced performance of the rotor speed estimations is also reflected in the RMS of the residuals, which are reported in Table 4. It can be observed that the computational time of the entire ten-second simulation for the two estimators is quite similar. Therefore, the usage of the proposed TI method does not imply a significant computational increase. In fact, the average CPU time of the proposed approach to conduct topology processing is approximately $4 \cdot 10^{-5}$ s. Since the CPU time of the rotor speed estimator is on the order of milliseconds per time step, it is clear that the proposed methodology is compatible from a computational point of view with the rotor speed estimator. In this respect, it is worth remarking that the topology processing algorithms described in Section 1 are not suitable for DSE implementations since they have high computational burdens and cannot be coupled with fast estimators.

3.3.2. Short-circuit at bus 30 followed by the disconnection of transformer 2-30

The simulated event is a three-phase short-circuit with zero fault impedance that occurs at$t = 1$ s at the low voltage terminals of transformer 2-30. Subsequently, the transformer protection opens after 100 ms, provoking a topology change that affects the rotor speed estimation of generator 30. Due to the inclusion of the new bus 40, once the fault is cleared the generator remains connected to the network through transformer 30–40 and line 25–40.

The results of the online TI are reported in Fig. 6. It can be observed that the proposed method accurately tracks the topology configuration, misclassifying the TI only during the duration of the short-circuit. Immediately after the fault, the method recovers the correct topology configuration providing the output “0110” (branch of the transformer 2-30 open).

As in the previous study case, the topology change affects the rotor speed estimation. Before the clearing of the fault, the elements of the D^{+} matrix that affect the rotor speed estimation of generator 30 are (1,2), (1,30) and (1,40) [40], i.e., the relevant frequency measurements to conduct the estimation are those of buses 2, 30 and 40. However, once branch 2-30 is disconnected, the frequency measurements from bus 2 must no longer be taken into account. If the topology configuration is not properly detected and the frequency of bus 2 remains considered for the estimation, the rotor speed estimate will be biased.

These considerations can be used to interpret Fig. 7, which shows the simulated and the estimated rotor speed of the generator at bus 30. Again, the TI allows a correct estimation of the generator rotor speed as

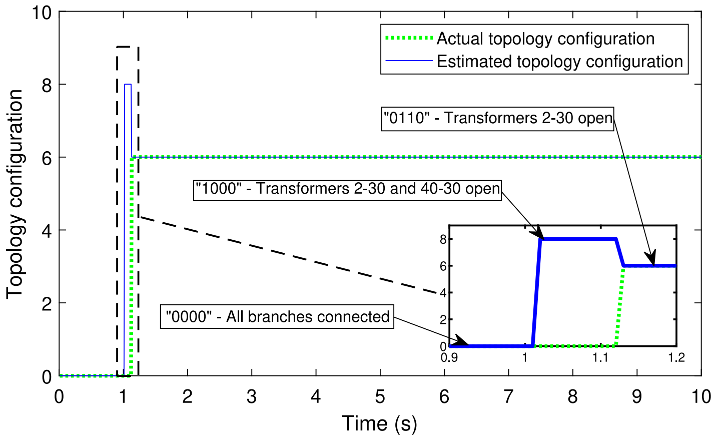


Fig. 6. Case 1-2 — Topology identification.

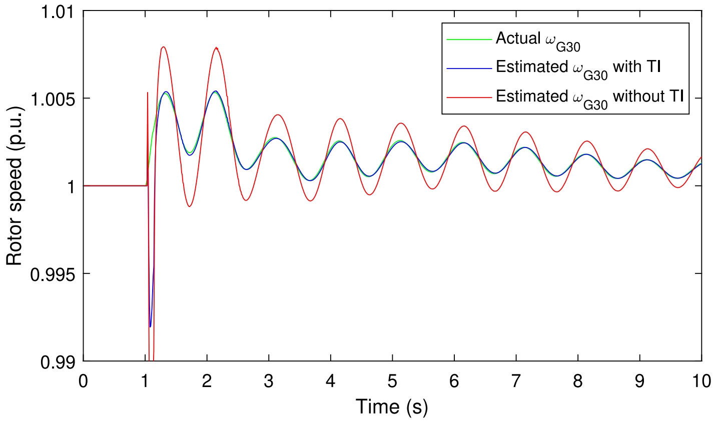


Fig. 7. Case 1-2 — Rotor speed estimation of synchronous machine 30.

Table 5
Case study 1-2.

Comparison index	Rotor speed estimation with TI	Rotor speed estimation without TI
$\Psi_{\omega_{G30}}$ (p.u.)	$8.613 \cdot 10^{-4}$	$1.764 \cdot 10^{-3}$
CPU time (s)	2.64	2.60

soon as the fault is cleared. Table 5 shows similar results to the previous case: the TI improves the accuracy in the speed estimation which results in a significant decrease of the RMS of the residuals, while the CPU time remains practically unchanged.

3.3.3. Additional remarks on the simulation results

The results reported in this section show that the proposed TI method is computationally compatible with the model-independent linear generator rotor speed estimator since the former has CPU times two orders of magnitude lower than those of the latter.

It is possible to observe that the higher the topology errors are, the greater the rotor speed estimation benefits from the TI. In fact, in the first case, the generator remains connected to the network through one of the two transformers connected in parallel. Hence, adjacent bus frequency measurements remain useful for estimating the generator rotor speed. Thus, the estimation bias is attributable to the wrong weights, i.e., the relative elements of D^{+}, given to the frequency measurements. In the second case, the complete disconnection of branch 2-30 implies that the bus 30 frequency measurements must not be taken into account once the switching device opens. Otherwise, the estimations are strongly biased. This justifies the significant difference between the estimation with and without the TI.

Finally, it is worth reminding also in this section that the proposed TI method can be coupled with any DSE algorithm that needs accurate and fast topology processing.

4. Study case 2: Decentralized application on the IEEE 118-bus test system

4.1. Training process and test results

The set of possible topology configurations increases rapidly with the size of the network, as more relevant branches are added to the system. Eventually, if the network is too large it is not possible to apply the proposed TI method to the whole system because the training process would be too time-consuming. However, the FD-based WLS estimator allows a decentralized estimation of the generator rotor speeds using only local PMU measurements. The proposed TI method can also be applied separately to each generator, using the same local PMU measurements and providing valuable information about the switching status of the branches connected to the generator bus.

This section demonstrates this decentralized approach by applying the proposed TI method and the FD-based WLS rotor speed estimator to generator 31 in the IEEE 118-bus test system. The application of the centralized method to the whole IEEE-118 Bus system would be difficult because of its relatively large size. Fig. 8 shows the portion of the system considered for this demonstration.

To train the DNN, a set of 4 PMU measurements is used, namely the voltage at bus 31 and the current at each branch connected to

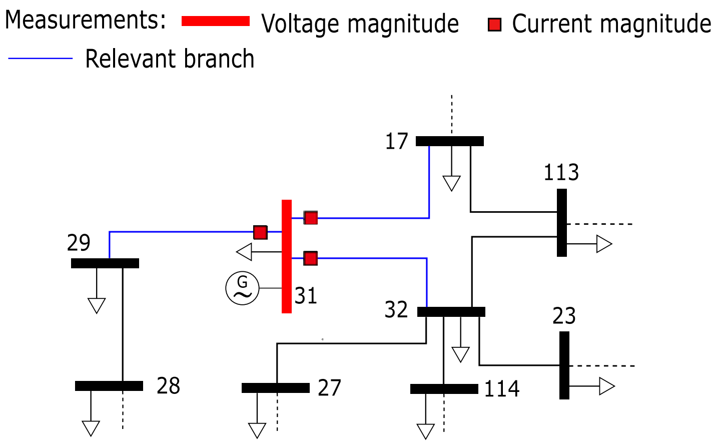


Fig. 8. Single-line diagram with the PMU used for TI and the relevant branches of the analyzed portion of the IEEE 118-bus test system.

Table 6
Training and test accuracy of the DNN decentralized approach.

	Decentralized approach DNN with 2 hidden layers 4-7-5-3
Training accuracy	99.98%
Training time	60 s
Testing accuracy	99.39%
Testing time	$3.9 \cdot 10^{-5}$ s

Table 7
Case study 2-1.

Comparison index	Rotor speed estimation with TI	Rotor speed estimation without TI
$\Psi_{\omega_{G31}}$ (p.u.)	$3.565 \cdot 10^{-4}$	$1.566 \cdot 10^{-3}$
CPU time (s)	1.84	1.81

bus 31. As in the previous case, the simulation time step is 10 ms and random Gaussian noise is added to the input measurements. Since the connection status of 3 branches must be considered, the number of relevant topology configurations for the rotor speed estimations of this generator is $2^3 = 8$. The set of topology configurations and their associated binary outputs, along with the input measurements, can be found in [40]. The learning rate used in this case is $lr = 10^{-4}$. As can be observed in Table 6, the DNN has 4 input neurons, 2 hidden layers of 7 and 5 neurons, and 3 output neurons. A total of 33 events are considered, and the training set has 2424 training samples.

4.2. Simulation results

4.2.1. Short-circuit at bus 31 followed by disconnection of lines 29–31 and 31–32

The simulation consists of a short-circuit at t=1 s located at bus 31, followed by the disconnection of lines 29–31 and 31-32 100 ms later. The disconnection of these two lines leads to a topological change that affects the estimation of the rotor speed of generator 31. For a proper estimation of the rotor speed, the frequency measurements from buses 29 and 32 should not be taken into account once the corresponding branches are opened.

Observing Fig. 9, it is possible to see how correct topology processing allows accurate rotor speed estimation. The RMS of the residuals and the CPU times of this case are reported in Table 7. As in the previous cases, it can be seen the TI improves significantly the accuracy of the estimation without practically affecting the computation time.

With respect to the previous study cases, the differences with the estimator that does not employ the TI are larger. This is because when no TI is employed, the estimator uses two frequency measurements, those of buses 29 and 32, that are no longer mathematically related to the generator 31 rotor speed in the WLS formulation.

4.2.2. Gross error in the measurement set

This subsection simulates the presence of a gross error in the measurement set. The events are the same as in Section 4.2.1: a short-circuit at bus 31 occurs at t=1 s and it is cleared by opening lines 29–31 and 31–32 100 ms later. In this case, a malfunction in a PMU input channel during the fault is considered; i.e., the current measurement I_{31-29} remains frozen at the short-circuit value of t=1.03 s for the entire simulation. The presence of this bad data in the TI input signals implies a wrong topology processing, which in turn translates into strongly biased rotor speed estimations.

As described in Section 2.3, to avoid this error a DNN-based BDDIS algorithm is implemented. The training set is created considering generation, load, and short-circuit events under all the relevant topology configurations of this case [40]. After different attempts, the DNN configuration that provides the best test results has 4 input and output neurons (number of measurements used in this case), and 2 hidden layers of 12 and 8 neurons, respectively. Due to similar reasons to the ones explained in Section 2.2, all the BDDIS DNN neurons employ the sigmoid activation function. The bad data identification threshold applied in this work is $\gamma_i = 10 \cdot \sigma_i$, where σ_i is the i_{th} measurement standard deviation. The learning rate used in this case is $lr = 4 \cdot 10^{-4}$, and the training time is approximately 25 min. As shown in Fig. 10, the presence of the BDDIS algorithm allows estimating a value of I_{31-29} that is similar to the true value of 0 p.u. (after the fault clearance), instead of the wrong value of 5.83 p.u. It is worth mentioning that the CPU time of the DNN-based BDDIS algorithm is $2.1 \cdot 10^{-5}$ s, which is of the same order of magnitude as the proposed TI method. Therefore, the BDDIS and the proposed TI methods are both suitable for DSE applications. In Fig. 11, it is possible to observe how the replacement of the wrong measurement allows the proposed TI to correctly estimate the right binary topology configuration “110” (lines 29–31 and 31–32 open) instead of output “101” (lines 17–31 and 31–32 open) that the TI provides in case no BDDIS is considered. The correct topology processing, in turn, implies a better estimation of the generator 31 rotor speed, as shown in Fig. 12.

5. Conclusion

The proposed TI method is able to detect topology changes in the analyzed cases, even immediately after a severe fault. The method shows a remarkably low computational load, taking approximately 40 μs to complete on the modified version of the New England test system using an average personal computer (PC). This makes the

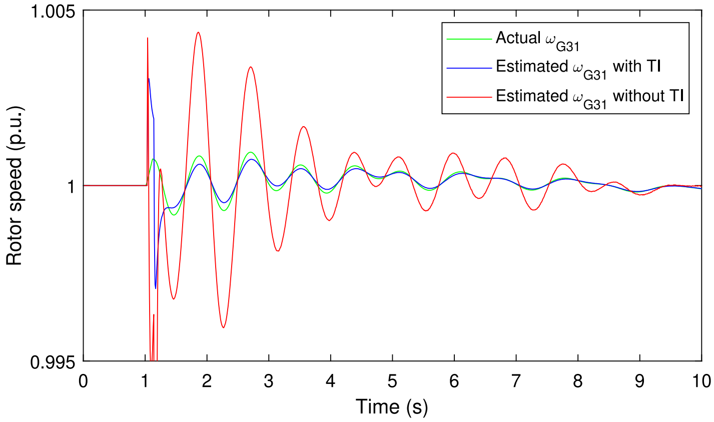


Fig. 9. Case 2-1 — Rotor speed estimation of synchronous machine 31.

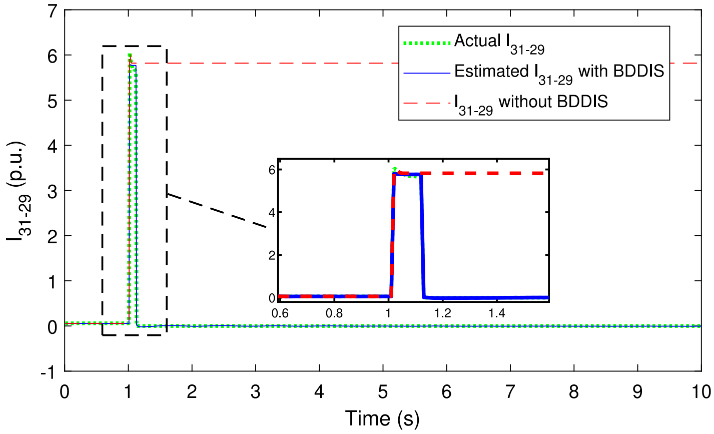


Fig. 10. Case 2-2 — I 31-29 estimation with and without BDDIS.

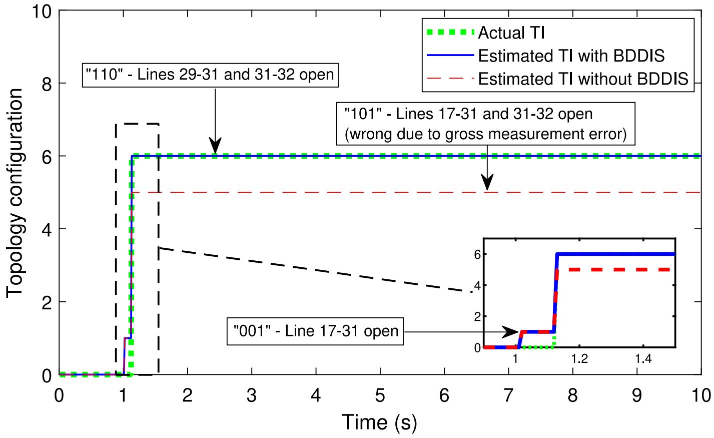


Fig. 11. Case 2-2 — Topology identification with and without BDDIS.

TI method suitable for fast online applications, as demonstrated by coupling it with a FD-based rotor speed estimator. However, the applicability of the proposed TI is not limited to this specific rotor speed estimator, and can also be applied to other DSE algorithms that rely on correct topology information.

A centralized implementation on a modified New England test system is proposed. This approach is straightforward since only one DNN needs to be tuned and trained, but it cannot be easily implemented on large electric networks. When applied to sizable systems, the main difficulty of the proposed algorithm lies on the training process. In the case of the centralized approach, with 10 generators and a set of 16 possible topology configurations, the proposed neural network takes 23 min to train on an average PC. An additional limitation of the centralized approach is that the optimization of the number of nodes in the DNN requires a previous evaluation by trial-an-error; too few nodes may cause inaccuracies while too many may not improve the performance. The decentralized approach solves these difficulties by splitting the topology identification into several smaller problems.

The decentralized application is demonstrated on a single generator of the IEEE 118-bus test system. This approach allows to avoid complex DNN training processes, although it requires the creation of one DNN per each device to be monitored. The number of PMU measurements

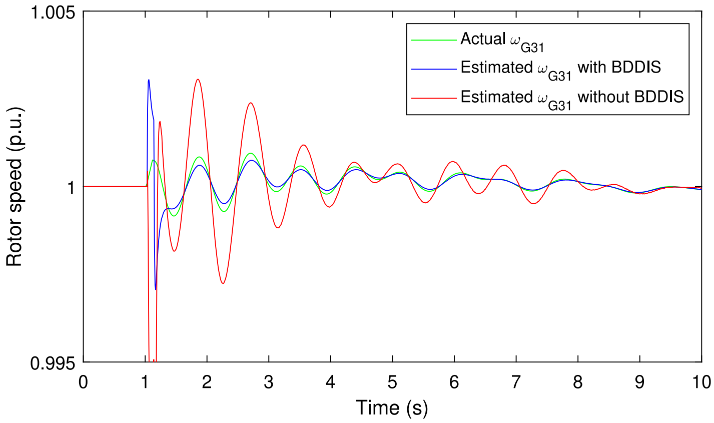


Fig. 12. Case 2-2 — Rotor speed estimation of synchronous machine 31 with and without BDDIS.

and possible topology configurations decrease, and the training process is simplified and reduced to 1 min. After testing all possible topology configurations, problems of local converge were not found neither in the centralized nor in the decentralized approach. In this study, the decentralized approach is coupled with a FD-based rotor speed estimator. However, it is worth reminding that other dynamic state estimators can benefit from this implementation.

The fact that neural networks act as black boxes that cannot be interpreted can be a limiting factor for their application to power systems, because under certain circumstances system operators may need to explain their actions to market regulators. For this reason, future research on the proposed method should be directed towards online applications in which other methods are not available because speed is crucial, for example wide-area protection schemes to improve transient and frequency stability. Future work is also needed to better understand the effect of measurement noise on the accuracy of the TI, specially when the noise follows a non-Gaussian distribution.

CRediT authorship contribution statement

Davide Gotti: Conceptualization, Methodology, Investigation, Software, Validation, Formal Analysis, Writing – original draft. **Pablo Ledesma:** Methodology, Supervision, Writing - review & editing. **Hortensia Amaris:** Supervision, Writing – review & editing.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgments

This work was funded by Agencia Estatal de Investigación MCIN/ AEI/ 10.13039/501100011033 under Grant PID2019-104449RB-I00.

References

[1] Zhao J, Gómez-Expósito A, Netto M, Mili L, Abur A, Terzija V, et al. Power system dynamic state estimation: Motivations, definitions, methodologies, and future work. IEEE Trans Power Syst 2019;34(4):3188–98. http://dx.doi.org/10.1109/TPWRS.2019.2894769.

[2] Zhao J, Netto M, Huang Z, Yu SS, Gomez-Exposito A, Wang S, et al. Roles of dynamic state estimation in power system modeling, monitoring and operation. IEEE Trans Power Syst 2021;36(3):2462–72. http://dx.doi.org/10.1109/TPWRS.2020.3028047.

[3] De La Ree J, Centeno V, Thorp JS, Phadke AG. Synchronized phasor measurement applications in power systems. IEEE Trans Smart Grid 2010;1(1):20–7. http://dx.doi.org/10.1109/TSG.2010.2044815.

[4] Terzija V, Valverde G, Cai D, Regulski P, Madani V, Fitch J, et al. Wide-area monitoring, protection, and control of future electric power networks. Proc IEEE 2011;99(1):80–93. http://dx.doi.org/10.1109/JPROC.2010.2060450.

[5] Liu Y, Singh AK, Zhao J, Meliopoulos AP, Pal B, Ariff MABM, et al. Dynamic state estimation for power system control and protection IEEE task force on power system dynamic state and parameter estimation. IEEE Trans Power Syst 2021;36(6):5909–21. http://dx.doi.org/10.1109/TPWRS.2021.3079395.

[6] Cui Y, Kavasseri RG, Brahma SM. Dynamic state estimation assisted out-of-step detection for generators using angular difference. IEEE Trans Power Deliv 2017;32(3):1441–9. http://dx.doi.org/10.1109/TPWRD.2016.2615594.

[7] Farantatos E, Huang R, Cokkinides GJ, Meliopoulos AP. A predictive generator out-of-step protection and transient stability monitoring scheme enabled by a distributed dynamic state estimator. IEEE Trans Power Deliv 2016;31(4):1826–35. http://dx.doi.org/10.1109/TPWRD.2015.2512268.

[8] Milano F. Rotor speed-free estimation of the frequency of the center of inertia. IEEE Trans Power Syst 2018;33(1):1153–5. http://dx.doi.org/10.1109/TPWRS.2017.2750423.

[9] Milano F, Ortega A, Conejo AJ. Model-agnostic linear estimation of generator rotor speeds based on phasor measurement units. IEEE Trans Power Syst 2018;33(6):7258–68. http://dx.doi.org/10.1109/TPWRS.2018.2846737.

[10] Milano F, Ortega A. A method for evaluating frequency regulation in an electrical grid - Part I: Theory. IEEE Trans Power Syst 2021;36(1):183–93. http://dx.doi.org/10.1109/TPWRS.2020.3007847.

[11] Ortega A, Milano F. A method for evaluating frequency regulation in an electrical grid - Part II: Applications to non-synchronous devices. IEEE Trans Power Syst 2021;36(1):194–203. http://dx.doi.org/10.1109/TPWRS.2020.3007851.

[12] Liu M, Chen J, Milano F. On-line inertia estimation for synchronous and non-synchronous devices. IEEE Trans Power Syst 2021;36(3):2693–701. http://dx.doi.org/10.1109/TPWRS.2020.3037265.

[13] Abur A, Gómez-Expósito A. Power system state estimation : theory and implementation. New York, NY, USA: CRC Press; 2004, p. 327.

[14] Lateef O, Harley RG, Habetler TG. Bus admittance matrix estimation using phasor measurements. In: 2019 IEEE power and energy society innovative smart grid technologies conference. Institute of Electrical and Electronics Engineers Inc.; 2019, http://dx.doi.org/10.1109/ISGT.2019.8791637.

[15] Korres GN, Manousakis NM. A state estimation algorithm for monitoring topology changes in distribution systems. In: IEEE power and energy society general meeting. 2012, http://dx.doi.org/10.1109/PESGM.2012.6345126, San Diego, CA, USA.

[16] Lourenço EM, Coelho EP, Pal BC. Topology error and bad data processing in generalized state estimation. IEEE Trans Power Syst 2015;30(6):3190–200. http://dx.doi.org/10.1109/TPWRS.2014.2379512.

[17] Vinod Kumar DM, Srivastava SC, Shah S, Mathur S. Topology processing and static state estimation using artificial neural networks. IEE Proc: Gener Transm Distrib 1996;143(1):99–105. http://dx.doi.org/10.1049/ip-gtd:19960050.

[18] Souza JC, Da Silva AM, Da Silva AP. Online topology determination and bad data suppression in power system operation using artificial Neural Networks. IEEE Power Eng Rev 1997;17(12):57. http://dx.doi.org/10.1109/pica.1997.599375.

[19] Zhao L, Liu Y, Zhao J, Zhang Y, Xu L, Xiang Y, et al. Robust PCA-deep belief network surrogate model for distribution system topology identification with DERs. Int J Electr Power Energy Syst 2021;125:106441. http://dx.doi.org/10.1016/J.IJEPES.2020.106441.

[20] Shadi MR, Ameli MT, Azad S. A real-time hierarchical framework for fault detection, classification, and location in power systems using PMUs data and deep learning. Int J Electr Power Energy Syst 2022;134:107399. http://dx.doi.org/10.1016/J.IJEPES.2021.107399.

[21] Gotti D, Amaris H, Ledesma P. A deep neural network approach for online topology identification in state estimation. IEEE Trans Power Syst 2021;36(6):5824–33. http://dx.doi.org/10.1109/TPWRS.2021.3076671.

[22] Azimian B, Biswas RS, Moshtagh S, Pal A, Tong L, Dasarathy G. State and topology estimation for unobservable distribution systems using deep neural networks. IEEE Trans Instrum Meas 2022;71:1–14. http://dx.doi.org/10.1109/TIM.2022.3167722.

[23] Singh D, Pandey JP, Chauhan DS. Topology identification, bad data processing, and state estimation using fuzzy pattern matching. IEEE Trans Power Syst 2005;20(3):1570–9. http://dx.doi.org/10.1109/TPWRS.2005.852086.

[24] Hayes B, Escalera A, Prodanovic M. Event-triggered topology identification for state estimation in active distribution networks. In: IEEE PES innovative smart grid technologies conference Europe. IEEE Computer Society; 2016, http://dx.doi.org/10.1109/ISGTEurope.2016.7856295, Ljubljana, Slovenia.

[25] Zhang J, Wang Y, Weng Y, Zhang N. Topology identification and line parameter estimation for non-PMU distribution network: A numerical method. IEEE Trans Smart Grid 2020;11(5):4440–53. http://dx.doi.org/10.1109/TSG.2020.2979368.

[26] Ma L, Wang L, Liu Z. Topology identification of distribution networks using a split-EM based data-driven approach. IEEE Trans Power Syst 2021. http://dx.doi.org/10.1109/TPWRS.2021.3119649.

[27] Liang D, Zeng L, Chiang HD, Wang S. Power flow matching-based topology identification of medium-voltage distribution networks via AMI measurements. Int J Electr Power Energy Syst 2021;130:106938. http://dx.doi.org/10.1016/J.IJEPES.2021.106938.

[28] Anagnostou G, Pal BC. Derivative-free Kalman filtering based approaches to dynamic state estimation for power systems with unknown inputs. IEEE Trans Power Syst 2018;33(1):116–30. http://dx.doi.org/10.1109/TPWRS.2017.2663107.

[29] Qi J, Sun K, Wang J, Liu H. Dynamic state estimation for multi-machine power system by unscented Kalman filter with enhanced numerical stability. IEEE Trans Smart Grid 2018;9(2):1184–96. http://dx.doi.org/10.1109/TSG.2016.2580584.

[30] Zhao J, Mili L. A decentralized H-infinity unscented Kalman filter for dynamic state estimation against uncertainties. IEEE Trans Smart Grid 2019;10(5):4870–80. http://dx.doi.org/10.1109/TSG.2018.2870327.

[31] Rouhani A, Abur A. Constrained iterated unscented Kalman filter for dynamic state and parameter estimation. IEEE Trans Power Syst 2018;33(3):2404–14. http://dx.doi.org/10.1109/TPWRS.2017.2764005.

[32] Zhao J, Mili L, Gómez-Expósito A. Constrained robust unscented Kalman filter for generalized dynamic state estimation. IEEE Trans Power Syst 2019;34(5):3637–46. http://dx.doi.org/10.1109/TPWRS.2019.2909000.

[33] Milano F, Ortega Á. Frequency divider. IEEE Trans Power Syst 2017;32(2):1493–501. http://dx.doi.org/10.1109/TPWRS.2016.2569563.

[34] Milano F, Ortega Á. Frequency variations in power systems: modeling, state estimation and control. Hoboken, NJ: Wiley-IEEE Press; 2020.

[35] NERC Reliability Guideline. PMU placement and installation. 2016.

[36] NERC Reliability Guideline. Power plant dynamic model verification using PMUs. 2016.

[37] Müller AC, Guido S. Introduction to machine learning with Python: a guide for data scientists. Sebastopol: O'Reilly Media, Incorporated; 2016.

[38] Haykin S. Neural networks: a comprehensive foundation. Hoboken, NJ, USA: Prentice Hall PTR; 1998, p. 842.

[39] Nielsen M. Neural Networks and Deep Learning. URL http://neuralnetworksanddeeplearning.com.

[40] Gotti D. Input measurements, output binary classification, D^+ elements, and PowerFactory source file of the test systems. 2022, http://dx.doi.org/10.13140/RG.2.2.14189.61925.

[41] Salehfar H, Zhao R. A neural network preestimation filter for bad-data detection and identification in power system state estimation. Electr Power Syst Res 1995;34(2):127–34. http://dx.doi.org/10.1016/0378-7796(95)00966-7.

[42] Sayghe A, Hu Y, Zografopoulos I, Liu X, Dutta RG, Jin Y, et al. Survey of machine learning methods for detecting false data injection attacks in power systems. IET Smart Grid 2020;3(5):581–95. http://dx.doi.org/10.1049/iet-stg.2020.0015, URL https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-stg.2020.0015.

[43] Prasanna Srinivasan V, Balasubadra K, Saravanan K, Arjun V, Malarkodi S. Multi label deep learning classification approach for false data injection attacks in smart grid. KSII Trans Internet Inform Syst 2021;15(6):2168–87. http://dx.doi.org/10.3837/tiis.2021.06.013.

[44] Ganjkhani M, Fallah SN, Badakhshan S, Shamshirband S, Chau K-w. A novel detection algorithm to identify false data injection attacks on power system state estimation. Energies 2019;12(11). http://dx.doi.org/10.3390/en12112209, URL https://www.mdpi.com/1996-1073/12/11/2209.

[45] Yang Z, Liu H, Bi T, Yang Q. Bad data detection algorithm for PMU based on spectral clustering. J Mod Power Syst Clean Energy 2020;8(3):473–83. http://dx.doi.org/10.35833/MPCE.2019.000457.

[46] Manitsas E, Singh R, Pal BC, Strbac G. Distribution system state estimation using an artificial neural network approach for pseudo measurement modeling. IEEE Trans Power Syst 2012;27(4):1888–96. http://dx.doi.org/10.1109/TPWRS.2012.2187804.

[47] DIgSILENT. PowerFactory 2021 user manual. Gomaringen, Germany; 2021.

[48] Pau M, Ponci F, Monti A, Sulis S, Muscas C, Pegoraro PA. An efficient and accurate solution for distribution system state estimation with multiarea architecture. IEEE Trans Instrum Meas 2017;66(5):910–9. http://dx.doi.org/10.1109/TIM.2016.2642598.

[49] Ahmad T, Senroy N. Statistical characterization of PMU error for robust WAMS based analytics. IEEE Trans Power Syst 2020;35(2):920–8. http://dx.doi.org/10.1109/TPWRS.2019.2939098.

[50] Bishop C. Neural networks for pattern recognition. USA: Oxford University Press; 1995.