An ELM-AE State Estimator for Real-Time Monitoring in Poorly Characterized Distribution Networks

P. N. Pereira Barbeiro, H. Teixeira
INESC TEC, Centre for Power and Energy Systems
Porto, Portugal
pnpb@inesctec.pt, hst@inesctec.pt

Jorge Pereira
INESC TEC and Faculty of Economics, University of Porto
Porto, Portugal
jpereira@inesctec.pt

R. Bessa
INESC TEC, Centre for Power and Energy Systems
Porto, Portugal
rbessa@inesctec.pt

***Abstract*—In this paper a Distribution State Estimator (DSE) tool suitable for real-time monitoring in poorly characterized low voltage networks is presented. An Autoencoder (AE) properly trained with Extreme Learning Machine (ELM) technique is the "brain" of the DSE. The estimation of system state variables, i.e., voltage magnitudes and phase angles is performed with an Evolutionary Particle Swarm Optimization (EPSO) algorithm that makes use of the already trained AE. By taking advantage of historical data and a very limited number of quasi real-time measurements, the presented approach turns possible monitoring networks where information of topology and parameters is not available. Results show improvements in terms of estimation accuracy and time performance when compared to other similar DSE tools that make use of the traditional back-propagation based algorithms for training execution.**

***Index Terms* - autoencoders, distribution state estimation, extreme learning machine, smart distribution networks.**

I. INTRODUCTION

The multitude of assets that are expected to be in operation in distribution grids of the future will contribute to increment significantly their complexity, resulting on several additional challenges to Distribution System Operators (DSO), especially regarding operational, security and reliability aspects. Loads, storage devices, generation units and electric vehicles supported by an advanced metering and communication infrastructure, capable of gathering and transmitting data in quasi real-time, could be a reality in the coming years. For the large majority of metering devices, it is predictable that all the acquired data might be recorded and transmitted to a database, where it will be stored for relatively long periods of time.

The existence of a new generation of SCADA systems located at the Distribution Management System (DMS) will be therefore indispensable. This system will assist DSO on the integration, control and management of the assets in such a coordinated way that the safe operation and power quality patterns could be guaranteed. The main goal is to monitoring and operating distribution systems in quasi real-time, giving to DSO a complete snapshot of their grids while outputs for other control modules are provided at the same time. For example, in the evolvDSO project [1], the Use Case methodology (IEC/PAS 62559) identified the state estimation function as a key element to increase grid observability.

The new SCADA system will rely on data that may be acquired by several different telemetry equipment dispersed among the grid such as Remote Terminal Units (RTU), Phasor Measurement Units (PMU), Intelligent Electronic Devices (IED) and Smart Meters (SM). These technologies can contribute decisively to diminish the lack of information in distribution grids. However, since monitoring all grid points in quasi real-time manner will be for sure economically unfeasible, a Distribution State Estimator (DSE) module will continue to be a mandatory tool in the future DMS (even on those existing in the most advanced smart distribution networks). Moreover, in the large majority of the distribution systems, there is a significant lack of information, particularly at Low Voltage (LV) level. This might be the most critical issue when it comes to the development of a DSE suitable to run in the future LV smart distribution systems. Therefore, in the context of this paper, it is assumed that a proper DSE must be prepared for dealing with the following two aspects:

i) networks where their topology and parameters are partially or completely unknown;

ii) full exploitation of all the sensorial information available, both in quasi real-time and historical.

Regarding the conventional state estimation techniques, widely used at the transmission level, it is well known that the key for their success relies on having a complete knowledge of the grid's technical parameters and topology (among other factors, such as a high level of redundancy of the quasi real-time measurements available). As it was stated before, this is not the case experimented in a large number of distribution grids, which turns the conventional state estimation techniques not suitable to be used. Several techniques have been proposed for well characterized distribution systems [2-4]. Despite most of them are able to dealing with the very particular characteristics of distribution systems (e.g. extremely large number of nodes and branches, unbalanced loads, large number of dispersed generation) and a few of them are even suitable to be used in smart distribution grids, all fail if tested for the requirement specified in **i)**.

Following a complete different approach, some authors

The research leading to this work is being carried out as a part of the evolvDSO project (*Development of methodologies and tools for new and evolving DSO roles for efficient DRES integration in distribution networks*) - www.evolvdso.eu. This project is funded by the European Commission under the seventh framework program (FP7) under grant agreement number 608 732.

have exploited the use of Artificial Neural Networks (ANN) and machine learning capabilities for recompose missing information in SCADA systems [5], identification of topology errors [6] and generation of pseudo-measurements [7]. In [8], an ANN-based hybrid state estimator is proposed for determining the state of the system in the presence of conventional asynchronous as well as synchronous PMU measurements. Analyzing the mentioned studies, one can conclude that few of them are focused on obtaining a state estimation solution and on the evaluation of its quality. Instead, the proposed methods were applied to perform other DSE related functions such as topological analysis, observability analysis, bad data detection, etc. Therefore, the referred studies have not been verified against the requirements specified in **i)** and **ii)**. Two recent works [9, 10] may be taken as an exception. In [9] the authors tackle both the issues **i)** and **ii)** through the use of autoencoders (AE) trained with a Resilient Back-Propagation algorithm (RPROP) [11] and then applying a meta-heuristic procedure for finding system state variables, i.e., voltage magnitudes and phase angles. A similar methodology was used in [10] but applied to the problem of three-phase state estimation in unbalanced distribution grids, both in MV and LV networks.

In the view of the above, this paper exploits the concept of Extreme Learning Machine Autoencoder (ELM-AE), which consists on the application of ELM techniques [12] to properly train an AE. This already trained AE can be seen as the "brain" of the DSE algorithm proposed. The main motivation behind this study is trying to exploit ELM features in order to improve estimation accuracy, training and running times of a DSE based on artificial intelligence concepts. The proposed DSE should satisfy the requirements specified in **i)** and **ii)** and to be proper to run in several different distribution systems configurations, namely when smart grid features are present. In addition to the system state variables, the usual output of a state estimator, active/reactive power injections at customers' place may be also estimated if required. The proposed DSE algorithm is evaluated against the one proposed in [9] for distinct real-time telemetry scenarios. Several performance indicators are used to evaluate the training and running phases in what concerns to time and accuracy.

II. METHODOLOGY

A. Distribution State Estimator Model

As it was early mentioned, the state estimator algorithm developed is based on the use of a specific type of ANN, the AE. Basically, AE are feedforward neural networks that are built to mirror the input space in their output, being the size of its output layer always the same as the size of its input layer. Therefore, an AE is trained to display an output equal to its input. Once the autoencoder is trained, if an incomplete pattern is presented, the missing components may be replaced by random values producing a significant mismatch between input and output. A search may then be conducted by an optimization algorithm to discover the values that should be introduced in the missing components such that the input-output error becomes minimized. More information about AE and their applications can be found in [5], [6] and [9].

The proposed DSE model is depicted in Fig. 1. As it can be seen, it was adopted an AE with only a single hidden layer. In the model it was also considered the existence of two distinct execution phases. The first phase (marked in the Fig. 1 as "DSE Training") corresponds to the train of the AE, while the second phase (marked in the same figure as "Running DSE") corresponds to the DSE quasi real-time execution.

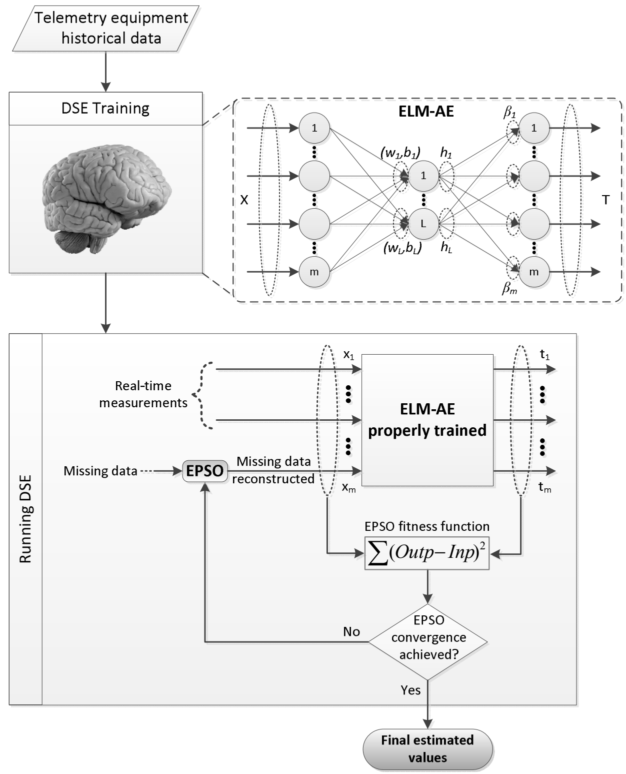

Figure 1. Model of the proposed DSE.

Regarding to the training phase, unlike conventional AE that usually apply back-propagation based algorithms for training purposes, here it is employed an ELM technique as the training algorithm for the AE (see section II.B.). Before training the AE, and in order to pre-treating the input and output training dataset, a standardization procedure must be run. This is a simple scale adjustment process, where the range of the input and output values are adjusted to a normalized interval [-1, 1], which fits the input variables to the range of the activation function.

In relation to the running process, an Evolutionary Particle Swarm Optimization (EPSO) algorithm makes use of the AE already trained to estimate the missing signals (see Fig. 1), which in the context of this work are the voltage magnitudes and phase values. Nevertheless, any other electrical variable can be included if desired. The only mandatory requirement is the existence of historical data related to them (e.g., active/reactive power quantities). The number and type of variables to be estimated will depend on their availability on the historical dataset as well as on the amount and type of measurements being telemetered in quasi real-time. For instance, at LV levels it is predictable that each customer owns a SM capable of measure voltage magnitude and active/reactive powers while the capability of measuring angles is not expected. Additionally, voltage magnitude may be enough for the functionalities expected to be used in this voltage level. On the opposite side, it is expectable measurements of voltage angles at MV level (acquired, for instance, through PMU).

It is important to mention that, in order to have an effective prediction of the system state for a given operating point, the measurements of all the electrical quantities existing in the historical database as well as the ones being transmitted in quasi real-time should be temporally synchronized.

B. Implementation of Extreme Learning Machine Concepts

Compared to the traditional gradient-descent based algorithms, particularly back-propagation, ELM [12] is a relatively recent computational intelligence technique that has been applied in different problems (e.g., regression, binary and multiclass classifications) with very promising results, both in terms of accuracy and computational performance [13]. Differently from the traditional training techniques, ELM is a non-iterative learning algorithm that only computes the weights vector between the hidden layer and the output layer, while the input weights matrix and the hidden layer biases vector are randomly generated (using, for instance, a uniform distribution) without tuning and independently from the input data [12, 14]. Thus, as these parameters can be fixed (are not required to be tuned), the output weights can be analytically calculated by solving a linear system of equations using the least-squares method. This is the essence of ELM techniques, contrary to the common understanding of learning. It has been shown that even considering that the input weights and hidden layer biases remain fixed after randomly generated, a single-layer neural network trained with ELM algorithm can maintain its universal approximation capability [14, 15]. Furthermore, and in contrast to the commonly used back-propagation learning algorithm which only minimizes the training error (without considering the magnitude of the weights), ELM minimizes not only the training error but also the norm of the output weights. ELM also offers some advantages such as fast learning speed, minimal human intervene and ease of implementation [16].

Mathematically, the concept of an ELM-AE is quite similar to the traditional ELM presented next. The input data is mapped from the m-dimensional input space to L-dimensional hidden layer feature space and the network output is given by

$$f_L(x) = \sum_{i=1}^{L} \beta_i h_i(x) = \mathbf{h}(x)\boldsymbol{\beta} \quad (1)$$

where $\boldsymbol{\beta} = [\beta_1, ..., \beta_L]^T$ is the output weight matrix between the hidden nodes and the output nodes, $\mathbf{h}(\mathbf{x}) = [h_1(\mathbf{x}), ..., h_L(\mathbf{x})]$ are the hidden node outputs with respect to the input $\mathbf{x}$ and $h_i(\mathbf{x})$ is the output of the i-th hidden node. Given N training samples $\{(\mathbf{x}_i, \mathbf{t}_i)\}_{i=1}^{N}$, where input data $\mathbf{x}_i = [x_1, ..., x_m]^T$ and the target $\mathbf{t}_i = [t_1, ..., t_m]^T$, the learning problem to be resolved can be compactly formulated as

$$\mathbf{H}\boldsymbol{\beta} = \mathbf{T} \quad (2)$$

where $\mathbf{T} = [\mathbf{t}_1, ..., \mathbf{t}_N]^T$ consists of the target matrix and $\mathbf{H} = [\mathbf{h}^T(\mathbf{x}_1), ..., \mathbf{h}^T(\mathbf{x}_N)]^T$ is the hidden layer output matrix. The i-th column of $\mathbf{H}$ is the i-th hidden node output with respect to inputs $\mathbf{x}_1, \ldots, \mathbf{x}_N$. The training procedure is equivalent to finding a least-squares solution $\hat{\boldsymbol{\beta}}$ of the linear system defined in (2) as

$$\left\| \mathbf{H}\hat{\boldsymbol{\beta}} - \mathbf{T} \right\| = \min_{\beta} \left\| \mathbf{H}\boldsymbol{\beta} - \mathbf{T} \right\| \quad (3)$$

If the number N of training samples is equal to the number L of hidden nodes, matrix $\mathbf{H}$ is square and invertible and the neural network can approximate these training samples with zero error [16]. In such case

$$\begin{aligned} \hat{\boldsymbol{\beta}} &= \mathbf{H}^{-1}\mathbf{T} \\ \boldsymbol{\beta}^T\boldsymbol{\beta} &= \mathbf{I} \end{aligned} \quad (4)$$

However, it is usual that the number N of training samples is different from the number L of hidden nodes. In this case, $\mathbf{H}$ is a nonsquare matrix and the smallest norm least-squares solution of the linear system defined in (2) can be calculated by

$$\hat{\boldsymbol{\beta}} = \mathbf{H}^{\dagger}\mathbf{T} \quad (5)$$

where $\mathbf{H}^{\dagger}$ represents the Moore-Penrose generalized inverse of matrix $\mathbf{H}$ and $\hat{\boldsymbol{\beta}}$ are the optimal output weights that minimize the training error.

To make the solution more robust and improve the generalization performance [17], a regularization term can be added [18]

$$\boldsymbol{\beta} = \left(\frac{\mathbf{I}}{C} + \mathbf{H}^T\mathbf{H} \right)^{-1} \mathbf{H}^T\mathbf{T} \quad (6)$$

$$\boldsymbol{\beta} = \mathbf{H}^T \left(\frac{\mathbf{I}}{C} + \mathbf{H}\mathbf{H}^T \right)^{-1} \mathbf{T} \quad (7)$$

where $\mathbf{I}$ is the identity matrix and C is a scale parameter. The choice between (6) or (7) depends on the dimension of the training dataset N and on the dimension of the feature space L. Both expressions can be used no matter the size of N and L, but computational costs are usually more reduced when using (6) for the cases where $N >> L$ and (7) for the cases where the training dataset is relatively small [18].

Basically, there are two main differences between the ELM-AE and the traditional ELM. The first one is that in an ELM-AE the target output $\mathbf{t}$ is the same as input $\mathbf{x}$ (as previously mentioned, an AE is trained to display an output equal to its input). The second one is that the input weights and the hidden nodes biases of an ELM-AE are made orthogonal after being randomly generated. Orthogonalization of these randomly generated parameters tends to improve the ELM-AE's generalization performance. As shown in [19], the orthogonal random weights and biases of the hidden nodes (that project the input data to a L-dimensional space) can be calculated as

$$\mathbf{h} = g(\mathbf{w} \cdot \mathbf{x} + \mathbf{b})$$
$$\mathbf{b}^T \mathbf{b} = 1 \tag{8}$$

where $\mathbf{w} = [\mathbf{w}_1,...,\mathbf{w}_L]$ are the orthogonal random input weights, $\mathbf{b} = [b_1,...,b_L]$ are the orthogonal random hidden layer biases and $g(.)$ is the activation function of the hidden nodes. Depending on the relation between the number m of input nodes and the number L of hidden nodes, orthogonality of input weights matrix may or not be complete, being verified in (8) as follows: when the number of input nodes is larger than the number of hidden nodes ($m > L$), $\mathbf{w}^T\mathbf{w} = \mathbf{I}$ is true; when the number of input nodes is smaller than the number of hidden nodes ($m < L$), $\mathbf{w}\mathbf{w}^T = \mathbf{I}$ is true; finally, when the number of input nodes coincides with the number of hidden nodes ($m = L$), $\mathbf{w}^T\mathbf{w} = \mathbf{w}\mathbf{w}^T = \mathbf{I}$ is true.

III. Case Study Description

All the data used in the simulations performed, including data related to the network, telemetry equipment, historical database, among other are completely characterized in [9]. This choice was made taking into account that one of the main goals of this work is to evaluate the proposed methodology by comparing it with the one presented in [9]. The most relevant data for the characterization of the case study is summarized in the next paragraph.

The considered network is a small typical Portuguese LV network of 33 nodes. The historical database consists on simulated data generated through power flows for a period of 4 months (summer season) in time steps of 15 minutes (11040 samples for training purposes and 672 samples for the running phase of the DSE). More details about the historical database creation can also be found in [9]. Regarding telemetry equipment, three scenarios were tested. In each scenario the number of SM (located at consumers' place) with capabilities of transmitting voltage magnitude and active and reactive power measurements in real-time was assumed to be different.

TABLE I. Quasi Real-time Measurements for each Scenario

Scenario	No of quasi real-time measurements (m)	Variables to be estimated (n)	m/n (%)
1	8	64	12.5
2	23	59	39.0
3	47	51	92.2

In Table I it is summarized, for each scenario, the number of quasi real-time measurements existing in the network (gathered from the metering equipment available in the network) and the total number of variables to be estimated. It is also computed in the same table the factor **m/n** that gives the relation between these two quantities (expressed in percentage).

IV. Case Study Results

In order to properly analyze the performance of the DSE model proposed under different perspectives, this section is divided in three subsections. The first section begins with the evaluation of the results obtained during the training stage regarding the generalization performance capability of the ELM-AE. Then, in the following two sections, accuracy and time performance are respectively evaluated for the case study presented in section III.

All the simulations were performed in a computer with an Intel Core i7-2600, 3.40 GHz CPU and 8 GB of RAM memory. In results presented, the bus voltage magnitudes and angles were considered as the state variables to be estimated. For each telemetry scenario evaluated, one week (last seven days from the historical database) was used as the test set for evaluate the proposed methodology. Since the amount and type of measurements present in the input dataset vary in each scenario (measurements available in quasi real-time and variables to be estimated), it was necessary to train three different ELM-AE, being the number of its input nodes equal to the number of measurements present in the input dataset. In all the simulations performed, it was used an ELM-AE with one hidden layer, where the number of hidden nodes assumed in each scenario was the same as in [9] (see Table IV). For activation function of the hidden nodes it was chosen a symmetric sigmoid function. The biases and input weights were generated through a uniform distribution between the normalization range [-1, 1]. In relation to the output nodes (the same number as the input), it was chosen a linear function as activation function. For comparison purposes, the convergence criterion adopted for the EPSO algorithm was a fixed number of 200 iterations (the same value as used in [9]). Other possibility could be to specify a given tolerance (error), and let the algorithm free to iterate until reach convergence.

A. Generalization Performance

The AE generalization performance is an important indicator evaluated in this work through the root mean squared error (RMSE) at the end of the training phase. Observing Table II, it is possible to see very significant differences between the two algorithms used for training the AE. Comparing the values of the RMSE for the testing set, it becomes clear that a better generalization performance is achieved with the ELM-AE. ELM-AE accounts for errors in the third decimal place while with the AE trained using a RPROP algorithm (hereafter named as RPROP-AE) errors occur in the first decimal place (in all the scenarios analyzed).

TABLE II. Generalization Performance

Training algorithm	Scenario	RMSE	
		Training Set	Testing Set
RPROP	1	0.4309	0.4330
	2	0.4435	0.4476
	3	0.6422	0.6430
ELM	1	0.0064	0.0062
	2	0.0079	0.0079
	3	0.0099	0.0105

B. DSE Accuracy Evaluation

In Fig. 2 are depicted boxplots for the voltage magnitude absolute error in all the network buses not being quasi real-time monitored. The absolute error was calculated between the real values (generated through power flows) and the estimated values obtained with the DSE model proposed. As it was expected, the estimation accuracy is improved when are available more measurements in quasi real-time. Similarly to the results presented in [9], the worst estimation occurs for buses that have both loads and microgeneration (e.g., buses number 17, 22, 25), what was expected due to the higher volatility of the power injected in these buses. In general, the error distribution shape obtained in both studies is analogous. However, the ELM-AE yields better results than the RPROP-AE. This can be easily verified observing the results shown in Table III, where are computed the mean absolute error (MAE) and the maximum absolute error (Max.) for the entire evaluation set considered. It is possible to see that the MAE obtained using the ELM-AE is nearly 50% smaller than the MAE attained using the RPROP-AE in scenario 2 and 3 and about 30% smaller in scenario 1.

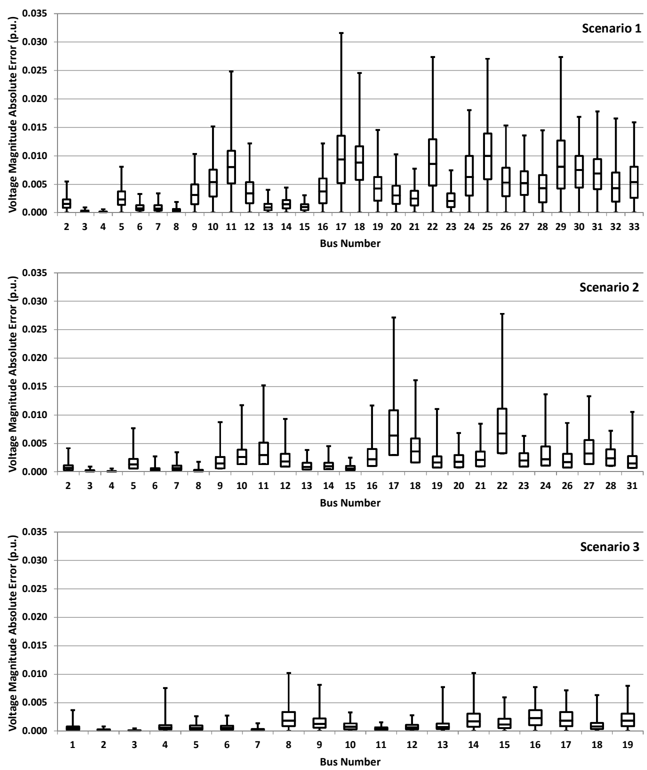

Figure 2. Voltage magnitude absolute error for all network buses (not being real-time monitored) in scenarios 1, 2 and 3.

Another aspect that evidence the better state estimation performed by the ELM-AE is the number of buses where the maximum absolute error remains below a threshold of 2%. While the RPROP-AE accounts for 11 and 6 buses violating the referred limit, respectively in scenario 1 and 2, the ELM-AE accounts only for 6 and 2 buses with such limit violations in the same scenarios. Taking these results into account and the way as the number of SM with capabilities of transmitting data in quasi-real time was defined in scenario 3 [9], the ELM-AE would probably requires less SM with such capabilities in scenario 3 for maintaining the error below the 2% threshold limit in all buses.

From Table III, it is possible to verify that the ELM-AE yields better results also for voltage angles, although the differences are not so obvious than in voltage magnitudes.

TABLE III. VOLTAGE ACCURACY

Training algorithm	Scenario	Magnitude (p.u.)		Angle (°)	
		MAE	Max.	MAE	Max.
RPROP	1	0.0062	0.0346	0.0780	0.5570
	2	0.0040	0.0327	0.0641	0.4940
	3	0.0020	0.0185	0.0480	0.4609
ELM	1	0.0044	0.0316	0.0688	0.5464
	2	0.0023	0.0278	0.0568	0.4629
	3	0.0011	0.0102	0.0405	0.2539

C. *Time Performance*

Table IV shows, for each scenario, a time performance comparison between the DSE model implemented with the ELM-AE and with the RPROP-AE. Before analyzing the results for the running times, one should bear in mind that they are strongly influenced by parameters such as the convergence criterion of the EPSO algorithm or the number of hidden nodes. In fact, a trade-off can be established between these parameters and the AE accuracy. Consequently, whenever running times is the most important issue, it may be reduced to the detriment of results accuracy (or vice versa).

Looking to Table IV it is possible to see that, for the same number of hidden nodes, the AE trained using ELM techniques is expressively faster than when a RPROP algorithm is employed. In average, the training process with the ELM algorithm runs nearly 2780 times faster than with RPROP algorithm. This may be explained due to the procedure employed in ELM-AE which is not an iterative learning process as in RPROP. Even assuming that, in the context of state estimation, the training phase may be done offline, these results might bring several advantages, especially when dealing with large amount of historical data and grids with hundreds of buses to be monitoring. In these circumstances a lot of time could be wasted on the adjustment of training parameters when using RPROP algorithm. Moreover, a fast training procedure may be a good feature in real networks with a lot of distributed renewable generation due to the big variability in load/generation patterns as well as changes in network configuration which may require performing DSE training more frequently. It should be noted that estimation accuracy tends to be improved when less different patterns exist in historical dataset, and logically if estimation is also performed under nearly similar conditions.

TABLE IV. TIME PERFORMANCE

Training algorithm	Scenario	Time (s)		No of hidden nodes
		Training	Running	
RPROP	1	335.109	1.274	43
	2	426.064	1.452	49
	3	657.778	1.673	59
ELM	1	0.093	1.345	43
	2	0.204	1.468	49
	3	0.249	1.555	59

In what concern to the DSE running time, the differences between both training algorithms are almost imperceptible. The ELM-AE accounts for slightly worse results in scenario 1 and 2. In scenario 3, where the number hidden nodes was higher, the results obtained with the ELM-AE were slightly better, which gives good indications that ELM may be faster than RPROP when the problem under study increases in size and complexity. On the other hand, the tiny differences verified may be mainly explained due to programming language differences. The DSE based on ELM-AE was totally coded in Python programming language, while the DSE based on RPROP-AE uses a neural network library coded in C language which was compiled through a *.dll* file for running the AE in Python environment (more efficient). On the other hand, if the EPSO convergence criterion used was settled by a tolerance value, it is reasonable to expect, looking to the generalization performance results (Table II), that the DSE based on ELM-AE could reach tolerance with less EPSO iterations than with the one based on RPROP-AE, resulting in lower execution times for the same accuracy degree.

V. Conclusion

In a paradigm of smart distribution grids it is expected that metering data may be stored and kept in a historical database at distribution management level. This historical information, together with some quasi real-time measurements gathered from terrain, may be the key to perform state estimation in poorly characterized distribution grids if properly exploited through the use of adequate artificial intelligence techniques. It should be kept in mind that in such conditions, conventional state estimation techniques cannot be applied due to the lack of knowledge about grid parameters.

In this paper, the concept of ELM-AE was tested as the "brain" of non-conventional DSE. The DSE model employed is similar to the one presented in [9], but instead of using a RPROP algorithm for training the AE it is based on the application of ELM concepts. The attempt was to improve the state estimation accuracy and reduce training and running times compared to the traditional training algorithms.

The results attained confirm one of the so referred advantages of the ELM techniques, the very low time needed to train the neural network. Compared to a RPROP-AE, the ELM-AE training time is nearly 2780 times faster. The execution time of the proposed DSE model follows close the values obtained for a DSE based on RPROP-AE, meaning that it is suitable for real-time applications. Nevertheless, the execution time has margin to be reduced if the algorithm is coded in a more efficient programing language.

In terms of state estimation accuracy, the indicators computed allow to conclude that ELM-AE performs better than RPROP-AE with error differences in the same order of magnitude. Anyway, the improvement registered for voltage magnitude is significantly higher than for voltage angle.

Other big advantage of ELM-AE is that the only parameter to be adjusted is the number of neurons in the hidden layer. Also, it does not require fine-tuning, since the input weights and the hidden layer biases can be fixed and the output weights can be determined analytically. Therefore, the optimum parameters may be found much easier than for back-propagation algorithm, where several training parameters need to be properly tuned.

As a final conclusion, it should be pointed out that the good generalization performance obtained with ELM gives indications that estimation error may be enhanced if a tighter convergence criterion for the EPSO algorithm is adopted.

References

[1] G. Bartolucci, *System Use Cases definition and functional and non-functional requirements for tools and methods, and definition of KPIs*, Deliverable D2.2, EU Project evolvDSO, Oct. 2014.

[2] C. N. Lu, J. H. Teng, and W. H. E. Liu, "Distribution system state estimation", *IEEE Trans. Power Systems,* vol. 10, no. 1, pp. 229-240, Feb. 1995.

[3] A. Ranković, B. M. Maksimović, and A. T. Sarić, "A three-phase state estimation in active distribution networks", *International Journal of Electrical Power & Energy Systems,* vol. 54, no. 0, pp. 154-162, 1//, Jan. 2014.

[4] G. N. Korres, N. D. Hatziargyriou, and P. J. Katsikas, "State estimation in Multi-Microgrids", *European Transactions on Electrical Power,* vol. 21, no. 2, pp. 1178-1199, Jun. 2011.

[5] V. Miranda, J. Krstulovic, H. Keko, C. Moreira, and J. Pereira, "Reconstructing Missing Data in State Estimation With Autoencoders", *IEEE Trans. Power Systems,* vol. 27, no. 2, pp. 604-611, May 2012.

[6] J. Krstulovic, V. Miranda, A. J. A. Simoes Costa, and J. Pereira, "Towards an Auto-Associative Topology State Estimator", *IEEE Trans. Power Systems,* vol. 28, no. 3, pp. 3311-3318, Aug. 2013.

[7] E. Manitsas, R. Singh, B. C. Pal, and G. Strbac, "Distribution System State Estimation Using an Artificial Neural Network Approach for Pseudo Measurement Modeling", *IEEE Trans. Power Systems,* vol. 27, no. 4, pp. 1888-1896, Nov. 2012.

[8] A. Kumar, and S. Chakrabarti, "ANN-based hybrid state estimation and enhanced visualization of power systems", in *Proc. 2011 IEEE PES Innovative Smart Grid Technologies*, pp. 78-83.

[9] P. N. P. Barbeiro, J. Krstulovic, H. Teixeira, J. Pereira, F. J. Soares, and J. P. Iria, "State estimation in distribution smart grids using autoencoders", in *Proc. 2014 IEEE International Power Engineering and Optimization Conf. (PEOCO)*, pp. 358-363.

[10] P. N. P. Barbeiro, H. Teixeira, J. Krstulovic, J. Pereira, and F. J. Soares, "Exploiting autoencoders for three-phase state estimation in unbalanced distributions grids", *Electric Power Systems Research,* vol. 123, pp. 108-118, 2015.

[11] M. Riedmiller, and H. Braun, "A direct adaptive method for faster backpropagation learning: the RPROP algorithm", in *Proc. 1993 IEEE International Conference on Neural Networks*, pp. 586-591.

[12] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications", *Neurocomputing,* vol. 70, no. 1–3, pp. 489-501, Dec. 2006.

[13] L.-C. Shi, and B.-L. Lu, "EEG-based vigilance estimation using extreme learning machines", *Neurocomputing,* vol. 102, pp. 135-143, Feb. 2013.

[14] G.-B. Huang, L. Chen, and C.-K. Siew, "Universal approximation using incremental constructive feedforward networks with random hidden nodes", *IEEE Trans. Neural Networks,* vol. 17, no. 4, pp. 879-892, Jul. 2006.

[15] R. Zhang, Y. Lan, G.-B. Huang, and Z.-B. Xu, "Universal Approximation of Extreme Learning Machine With Adaptive Growth of Hidden Nodes", *IEEE Trans. Neural Networks and Learning Systems,* vol. 23, no. 2, pp. 365-371, Feb. 2012.

[16] G.-B. Huang, D. Wang, and Y. Lan, "Extreme learning machines: a survey", *International Journal of Machine Learning and Cybernetics,* vol. 2, no. 2, pp. 107-122, Jun. 2011.

[17] L. L. C. Kasun, H. Zhou, G.-B. Huang, and C. M. Vong, "Representational Learning with ELMs for Big Data", *IEEE intelligent systems,* vol. 28, no. 6, pp. 31-34, 2013.

[18] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme Learning Machine for Regression and Multiclass Classification", *IEEE Trans. Systems, Man, and Cybernetics, Part B: Cybernetics,* vol. 42, no. 2, pp. 513-529, Apr. 2012.

[19] W. Johnson, and J. Lindenstrauss, "Extensions of Lipschitz Mappings into a Hilbert Space", in *Proc. 1984 Conf. Modern Analysis and Probability*, pp. 189-206.