

Deep Neural Network Based Distribution System State Estimation Using Hyperparameter Optimization

Gergő Békési, Lilla Barancsik, and Bálint Hartmann., *Member, IEEE*

This paper presents an innovative approach to optimizing deep neural network (DNN) hyperparameters for Distribution System State Estimation (DSSE). The authors highlight the rarity of DNN technology in real LV networks and emphasize the significance of their work. Compared to the previous study of the authors, this research shows improved estimations of voltage amplitudes and load angles with significantly reduced relative error (14-73%). The algorithm's non-iterative nature and the DNN architecture's ability to perform parallel computations with the help of GPU usage enabled a four order of magnitude improvement in running time, making it a useful tool for renewable energy integration planning and real-time monitoring.

Distribution System State Estimation, Deep Neural Networks, Hyperparameter Optimization, Tree-structured Parzen Estimator

I. INTRODUCTION

IN the past decade, state estimation (SE), more specifically distribution system state estimation (DSSE), has been a frequent topic of discussion in the international scientific community regarding power systems. The reason for the popularity and relevance of this topic is that, in order to sustainably maintain today's power industry, it is important to give special attention to the examination of distribution networks. With the decrease in the role of centralized energy production, these network segments are taking on a central role.

Primadianto and Lu [1] highlight the significance of DSSE in creating proper network models that can be used for real-time monitoring and analysis. They also point out the difficulties that arise in developing DSSE due to the transition to a sustainable energy environment that result in complex and fast aggregated generator and load dynamics, and the need for future research in the area. Ahmad et al. [2] highlight that with the advent of smart grid, the distribution grid is becoming more active and power flows are becoming bi-directional. This necessitates the development of DSSE for the successful implementation of various techniques such as protection, optimization, and control. The paper provides an overview of DSSE methods and identifies future research directions such as load forecasting and incorporating phasor measurement unit (PMU) or micro-PMU and smart meter data for the development of advanced distribution management systems. Majdoub et al. [3] conduct a comprehensive review of various DSSE methods that have been previously published in literature, including techniques based on adapting the Weighted Least Squares (WLS) algorithm from transmission to distribution network and methods based on machine learning and optimization algorithms, such as

Evolutionary Algorithms, Artificial Neural Network, Fuzzy Logic and Particle Swarm Optimization. The authors conclude that using DSSE methods based on advanced optimization method appears to be a promising solution for tackling complex nonlinear optimization problems. In particular, they note that the use of Deep Neural Networks (DNN) offers a reliable solution for overcoming observability issues.

In the field of DNNs Mestav et al. [4] perform deep learning simulations on Matpower test networks [5] with the support of Bayesian optimization for input generation. Results show the benefits of deep learning, particularly its robustness against bad data. Another solution is the DSSE performed with fixed DNN architectures and parameter optimization [6], which is tested on the IEEE 118-bus [7] and IEEE 14-bus [8] networks. It is shown in this article, that an adequate parameter optimization technique increases the precision by approx. 20-50% and reduces the computational time by approx. 30-50%. Zhang et al. [9] also tested DNN on the IEEE 118-bus network and achieved better results than the WLS algorithm. They optimize the hyperparameters of their linear layers using their own prox-linear tool by observing the results obtained.

In real-world applications of LV DSSE, the use of DNN is particularly rare. This technology has been utilized in the evoLVDSO and LV SCADA projects. The first one employs an DNN that has been trained on historical data to predict the voltage profile of an LV network through a state estimator tool [10]. While the second one creates an LV management system with an artificial intelligence-based state estimation tool to tackle typical DSSE problems [11].

In these projects, at least five or six PMU devices were used, which is necessary for achieving accurate DSSE results on real networks. DSSE requires access to real meter data that accurately reflects the conditions of the network and determining the precise location of the meters is a challenging task that requires careful consideration and planning. So, the successful application of DSSE on real networks depends on various factors, including the quality and availability of meter data, as well as the optimal positioning of the meters.

In reviewing the literature, it is apparent that, the application of modern hyperparameter optimization (HPO) techniques in low-voltage (LV) DSSE using neural network-based models is still poorly documented. Systematic grid optimization can be found in the literature in this field [12], or examples of Bayesian optimization with Gaussian processes fitted in predictive areas [13], but there is limited published research on detailed analysis of hyperparameters for DNNs specifically performing DSSE.

This work is intended to take a step towards filling this gap by demonstrating an DNN-based DSSE on real Hungarian LV networks using the Tree-structured Parzen Estimator algorithm, which is a Bayesian HPO method.

The rest of the paper is organized as follows. Section II. introduces the methods and data used for the work. Results are presented and discussed in Section III., and finally, comparisons and conclusions are drawn in Section IV. and V.

II. METHODS AND DATA

In this section, the properties of the networks modeled are first presented, followed by the presentation of the DNN input, output and model, and finally a description of the hyperparameter-optimizer Tree-structured Parzen Estimator (TPE) algorithm.

A. Networks

The experiments for this study were conducted in the same four real-life LV supply areas located in Hungary as presented in [14]. This work utilizes the previously established method for modeling LV networks that addresses the challenges of obtaining network topology information. It creates network models using data from the DSO information systems and addresses lines that have inconsistent or uncertain lengths. Each modeled area is comprised of an external grid element, a MV/LV transformer, and all the lines, buses, and loads on its LV side. The areas are identified by the numbers 18680, 44600, 44333, and 20667, and they each contain 2, 4, 5, and 10 circuits, respectively. More detailed information about the networks can be found in the original publication [14].

As emphasized in Section I., DNN-based DSSE is quite rare in the case of real networks and the unified automated application for multiple systems is unique. This standardization is thanks to TPE, which is considered new in this field and selects DNN characteristics to be adapted to the given system at a level of abstraction one step higher than that of the training.

B. DNN input and output

The neural network used in this work and described in detail in the II/C. subsection has a multi-layer perceptron architecture. The number of power inputs is double the number of consumers, and the number of outputs is double the number of nodes. The pseudo measurements of active and reactive power values are available for every 15-minute interval throughout the year. This database is created from a combination of standard load profiles (SLP) from Hungarian DSO, E.ON [15] and a sample of actual active power measurements.

One of the main advantages of the neural network solution is its ability to use non-electrical data as input as well. This allows temperature data [°C], sun altitude data [°], direct, scattered and reflected radiation data [kW/m²], wind [km/h] and precipitation data [mm] to be channeled from the Hungarian Meteorological Service database [16] and the Photovoltaic Geographical Information System of the Joint Research Centre [17] into the state estimator for the purpose of improving the estimations by recognizing the relationships between these data and consumption patterns. It is also essential to normalize these data

in order to achieve good learning performance [18].

The utilization of temporal data also has a great advantage, as the electricity consumption habits of individual consumers can vary in different ways from the SLP consumption at given locations. With this, the data is characterized in two different time perspectives:

- The first perspective is what time of day to examine within a given day. This means the 0:00 - 23:45 period with 15-minute increments.
- The second perspective is the day of the year. This means 365 data points from January 1 to December 31.

It is also important to note that these data cannot be treated as simple linear data. It is essential to provide the information to the neural network that 0:00 follows 23:45, and January follows December since such data may exhibit similarities. As a result, the input time variables are as follows:

$$x_{t1} = \sin(h \cdot \pi/12 + m \cdot \pi/720), \quad (1)$$

$$x_{t2} = \cos(h \cdot \pi/12 + m \cdot \pi/720), \quad (2)$$

$$x_{d1} = \sin(2 \cdot \pi \cdot d/365), \quad (3)$$

$$x_{d2} = \cos(2 \cdot \pi \cdot d/365), \quad (4)$$

where h is the hour of the day, m is the minute of the hour, and d is the day of the year.

The outputs of the SE are the voltage amplitudes and the load angles, which are also to be normalized separately. The method presented in this work uses the voltage amplitude values of 2 nodes as input for all four areas, where the voltage measurement devices are placed. The optimal meter-placement is performed by the TPE since it is an HPO task. Its operation is detailed in subsection II/D. The output of the load flow based on representative consumer data of a full year (52 weeks) serves as an appropriate benchmark to establish a virtual scenario, to which the neural network output is compared, and the difference (in the form of loss) is propagated back during training. Of these voltage data, the data from odd weeks are used for training and the data from even weeks are used for validation. Thus, the way the input SLP data and validation data are used is the same as in the authors' previous work [14], but due to the swap of the training and validation dataset, the DNN must perform even better than the new and better WLS result. The complete data management of the work is illustrated in Fig. 1. The calculation of the loss can occur between the two data sets marked with the dotted red line and thus the backpropagation represented by the red arrow can be realized. The loss calculation happens according to the p^{th} power of the absolute value of the difference between the datasets.

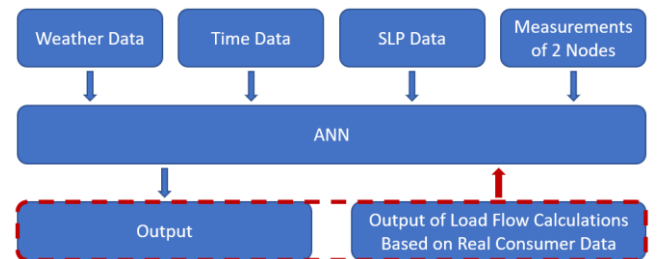


Fig. 1. Data management.

The representation of temporal data based on Equations (1-4) using a unit circle allows an DNN to leverage temporal continuity and relationships between individual time frames. Directly utilizing weather data in the computational graph is not unique, but it is also a new approach that can improve performance.

C. Architecture and Training of DNN

The building blocks of a neural network are the layers. Multiple layers perform computation between the network's input and output data, referred to as hidden layers except for the first and last layer. The output of the first layer can be described based on (5):

$$out_1 = z \cdot W_1 + b_1. \quad (5)$$

Where z is the vector of input data mentioned in subsection II/B., W_1 is the layer's weight matrix with the same number of columns as the number of input data and the same number of rows as the size of the next hidden layer, b_1 is the layer's additive term, a vector with the same number of elements as the size of the next hidden layer, and out_1 is the layer's output, a vector with the same number of elements as the size of the next hidden layer.

Then, further layers take the output of previous layers as input, with $ReLU$ [19] non-linearity according to equation (6):

$$out_{i+1} = ReLU(out_i) \cdot W_{i+1} + b_{i+1}. \quad (6)$$

Where out_i is the output of the i^{th} layer, a vector with the same number of elements as the layer size, W_{i+1} is the weight matrix of the $(i+1)^{th}$ layer, with the same number of columns as the output size of the i^{th} layer and the same number of rows as the size of the next layer or the output size in the case of the last layer, b_{i+1} is the additive term of the $(i+1)^{th}$ layer, a vector with the same number of elements as the size of the next layer or the output size in the case of the last layer, and out_{i+1} is the output of the layer, a vector with the same number of elements as the size of the next layer, or the output in the case of the last layer.

The training of the DNN is carried out by the widely used Adam [20] algorithm applying a learning rate (lr) decrease at every n^{th} step ($step\ size$) with a g ($gamma$) factor. The lr , $step\ size$, $gamma$, p and $batch\ size$ parameters, the number and size of the hidden layers and the optimal places for the measurement units are determined by the HPO algorithm detailed in the II/D. subsection.

The choice of the activation function and the definition of the hyperparameter ranges are critical in the development of a DNN. $ReLU$ is a popular choice for the activation function due to its computational efficiency and ability to handle the vanishing gradient problem, and each hyperparameter plays a crucial role in the learning process of a DNN. Finding the appropriate values for them is key to achieving high accuracy. Wrong configuration of the lr -related parameters can result in oscillations, overshooting of the optimal solution, or slow

convergence, while p affects the performance of the end of the training. The choice of $batch\ size$ represents a trade-off between convergence speed and stability, while the appropriate hyperparameter selection of the hidden layers can help balance the model's complexity and its ability to generalize. Therefore, these hyperparameters were selected for optimization.

D. The HPO of DNN

The TPE is developed to provide the best possible latency improvement by achieving an expected accuracy target in terms of the y loss function [21]. It uses the historical evaluations of the hyperparameter configurations (x) to create probabilistic models, which can suggest next set of hyperparameters for evaluations. TPE simulates $p(x|y)$ and $p(y)$ with Gaussian mixtures to indirectly obtain $p(y|x)$. According to (7), the I model improvement is created by choosing a new configuration:

$$I(x) = ReLU(y^* - y(x)), \quad (7)$$

where y^* is the threshold value of the loss function. The expected improvement is calculated by (8):

$$EI_{y^*}(x) = \int_{-\infty}^{y^*} (y^* - y) \frac{p(x|y) \cdot p(y)}{p(x)} dy. \quad (8)$$

$p(x|y)$ is modeled as $g(x)$ if $y \geq y^*$, and as $l(x)$ otherwise, where $g(x)$ is the probability density of the loss values more than y^* , and $l(x)$ is the probability density of the remaining values. By factoring out $p(x)$ from (8):

$$EI_{y^*}(x) = \frac{\int_{-\infty}^{y^*} (y^* - y) \cdot l(x) \cdot p(y) dy}{p(y \geq y^*) \cdot g(x) + (1 - p(y \geq y^*)) \cdot l(x)}, \quad (9)$$

which is equivalent to (10):

$$EI_{y^*}(x) = \frac{y^* - p(y \geq y^*) \cdot y^* - \int_{-\infty}^{y^*} p(y) dy}{1 - p(y \geq y^*) + p(y \geq y^*) \cdot \frac{g(x)}{l(x)}}, \quad (10)$$

TPE uses this last formula. It evaluates the hyperparameters iteratively by $g(x)/l(x)$ to find the hyperparameters x^* with the maximum expected improvement. In this work the *epoch number* is 100 for every evaluation, the first decile of evaluation loss values is used as a threshold value, and the whole process is started with 10 random evaluations, which is a common approach in the field of TPE [22].

III. RESULTS

This section demonstrates the accuracy improvement resulting from the neural network, likely due to the incorporation of extra temporal and weather data, and the significant speed increase due to the new technology and the easily usable GPU parallel computing technology.

The evaluation methodology for SE is the same for each transformer area. First, the results of optimization with TPE are presented, then, using visual color-based representation, the new neural network solution is compared with the previous

WLS technique. The TPE selects the hyperparameters in the simulation space where the DNN performs best for the state estimation, as described in the II/D. subsection. The hyperparameter ranges are shown on the diagrams of Fig. 2, 5, 8, and 11. The performance is determined by the magnitude of the average absolute error of the state variable with the maximum average absolute error over time. In the next subsections, the average absolute errors of the 18680, 44333, 44600 and 20667 transformer areas are presented respectively.

A. Results of supply area 18680

In the case of this smaller area, a smaller number of hidden layers is preferred due to its smaller complexity.

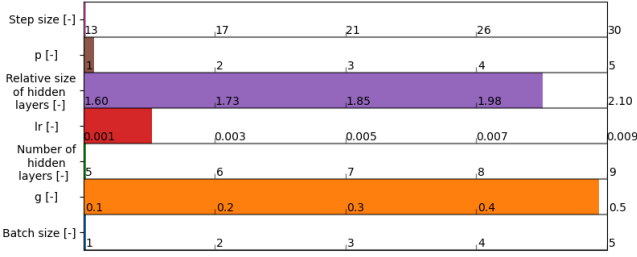


Fig. 2. Optimal hyperparameters of supply area 18680.

Similar to the later transformer areas, training is more effective with smaller batch size, but of course estimation can still be performed with larger batch sizes, even including the full dataset if substantial GPU support is available. The optimal size of hidden layers is clearly close to the number of output data, which, after examining other areas, is likely due to the smaller network size, since this is less characteristic in case of larger systems. A smaller learning rate is found to be more effective, but it comes with a slower decay rate and a smaller step size. The optimal value of p approaches 1, therefore a loss function similar to L1 leads to good results.

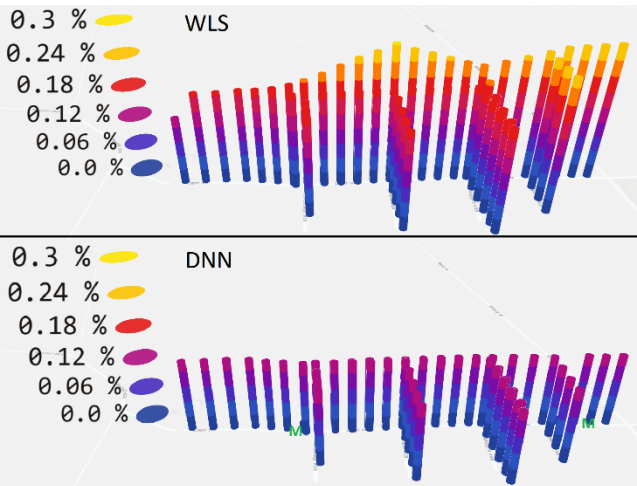


Fig. 3. Voltage amplitude errors of 18680.

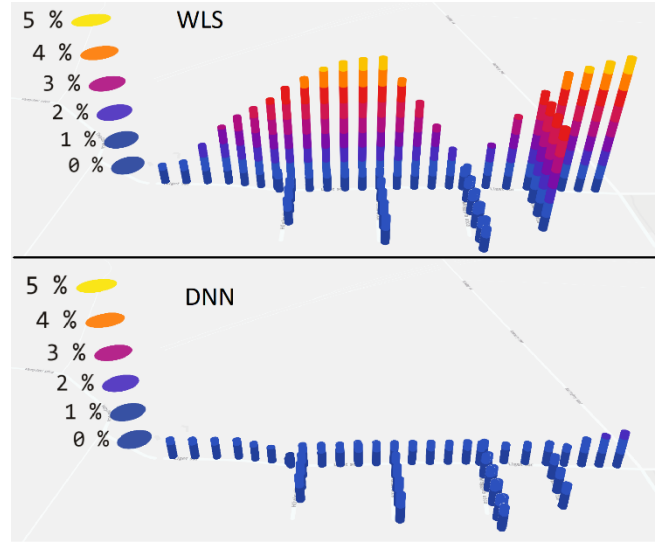


Fig. 4. Load angle errors of 18680.

The improvement compared to WLS in estimation results in this area is the most significant among all areas. The relative error of the node with the highest relative error decreases from 0.27% to 0.14% for the voltage amplitudes, and from 4.5% to 1.2% for the load angles. Even the maximum error is smaller in the case of the neural network solution than the average error of WLS. The speed increase, like in other areas, is significant here, being the second best among all areas in terms of ratio. The WLS takes 4620 seconds to perform the state estimation for the half-year data set, while the neural network takes 0.22 seconds. This means a 21,000-times faster performance, which can be a significant improvement in terms of efficiency for a network simulation team of a DSO.

B. Results of 44333

Area 44333 is similar to area 18680 in terms of the number and relative size of hidden layers and batch size. However, a larger p leads to better results with a larger learning rate, but this requires a faster decay and a larger step size. This larger p means a bigger sensitivity for outliers.

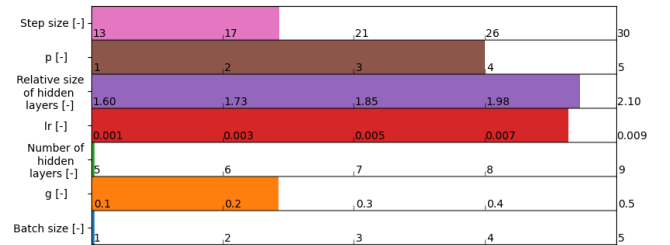


Fig. 5. Optimal hyperparameters of supply area 44333.

Once again, the DNN outperforms the WLS approach. The relative error of the maximum voltage amplitude and the relative error of the load angle at the nodes with the highest relative error decrease from 0.30% to 0.24% and from 4.5% to 2.8%, respectively. On average, these decreases mean changes from 0.22% to 0.16% and from 1.6% to 1.5%. The speed increase is the most significant in this area, as the estimation

can now be completed in 0.24 seconds instead of 7,550 seconds, thanks to the neural network.

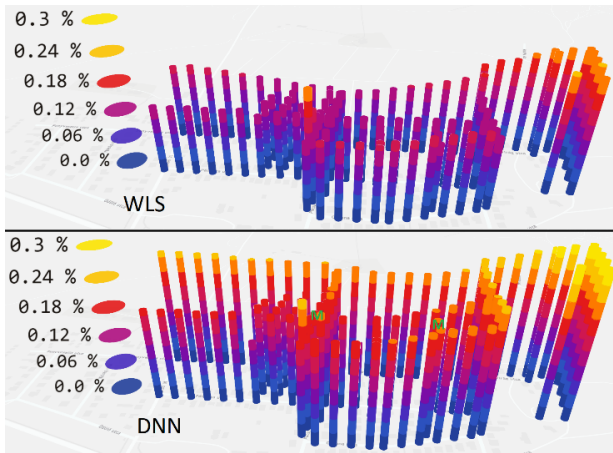


Fig. 6. Voltage amplitude errors of 44333.

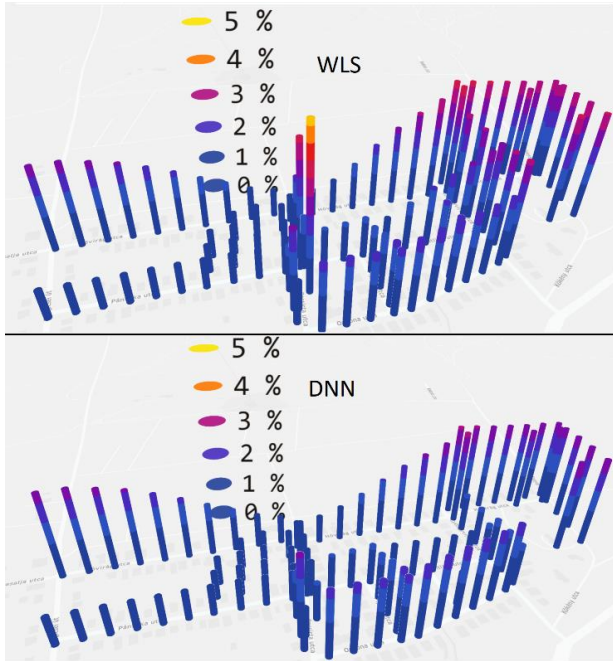


Fig. 7. Load angle errors of 44333.

C. Results of 44600

The batch size of 44600 is also 1, but it has more and smaller sized hidden layers. Its p value is similar to that of the smallest network, but it has a moderate weight decay and learning rate during its optimal training process with a larger step size. This smaller p means a smaller sensitivity for outliers.

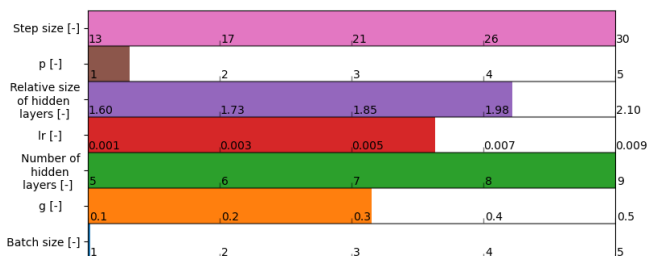


Fig. 8. Optimal hyperparameters for supply area 44600.

In this area, the DNN is better than the WLS in 4 success criteria values, while not changing 1 success criteria. The unchanged success criteria is the maximum relative error of the voltage amplitude, which is 0.26% for both the WLS and DNN. However, the maximum value improves from 5.0% to 4.3% in the case of the load angle, and the average improvements are modifications from 0.21% to 0.15% and from 3.2% to 2.2% in case of the voltage amplitude and load angle respectively. The decrease in runtime from 9,072 seconds to 0.72 seconds represents a speed increase of 12,600 times.

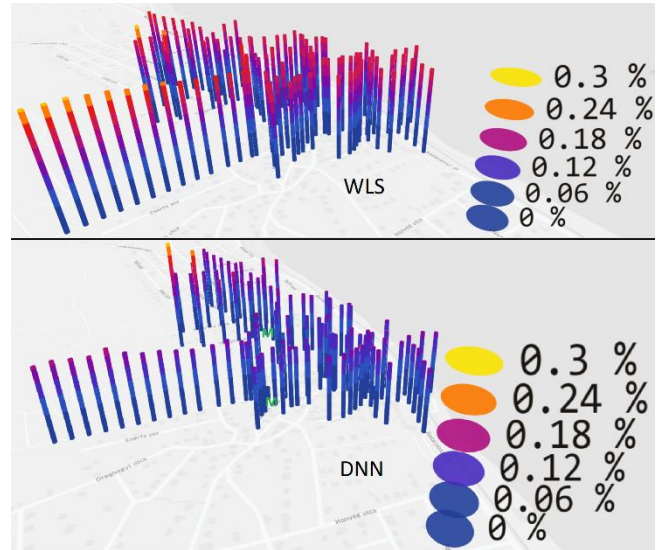


Fig. 9. Voltage amplitude errors of 44600.

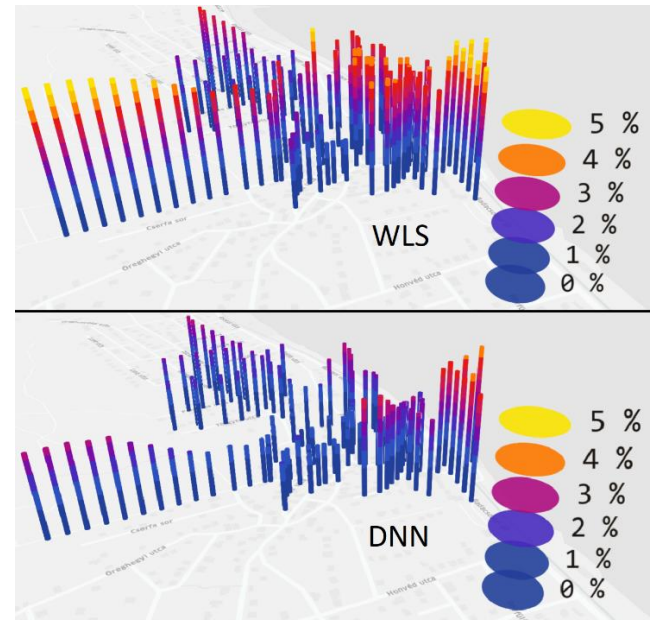


Fig. 10. Load angle errors of 44600.

D. Results of 20667

The area of 20667 is the largest of all. It has a small relative hidden layer size, but due to the network size it is considered large in absolute terms. It has a bit more layers, a medium sized p value, and bigger batch size can be applied. This area is

characterized by a big weight decay, like area 18680, but has a moderate learning rate, and near as big step size as area 44600.

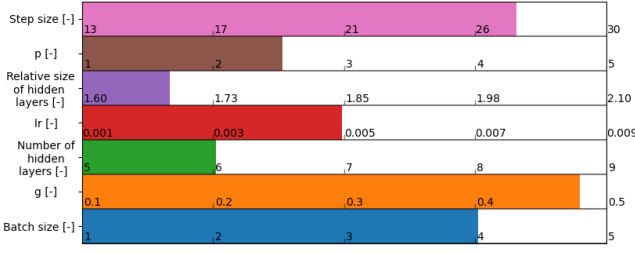


Fig. 11. Optimal hyperparameters for 20667.

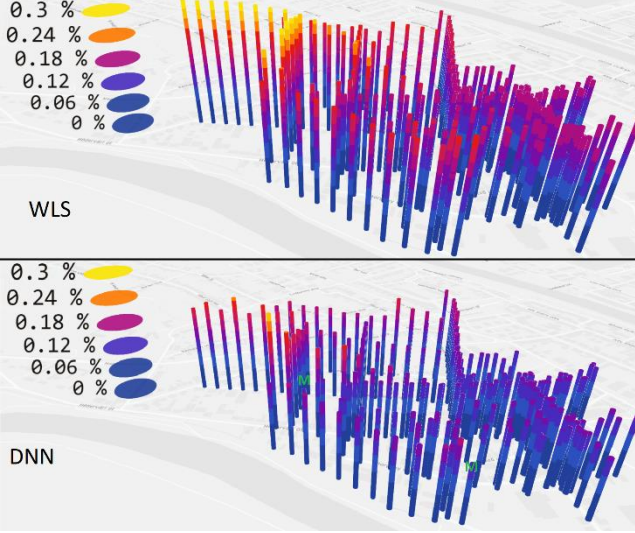


Fig. 12. Voltage amplitude errors of 20667.

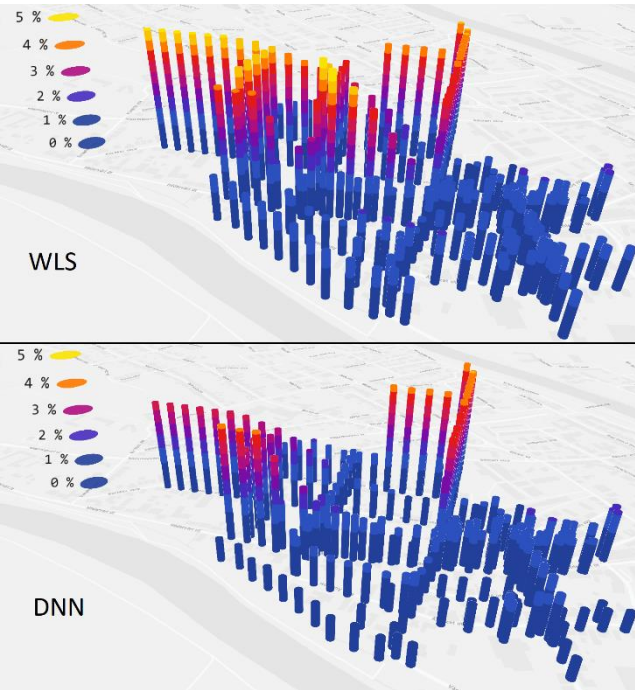


Fig. 13. Load angle errors of 20667.

The average load angle error is better when using the WLS technique, because the 1.4% error outperforms the 1.5% error.

However, all other metrics, including the average voltage amplitude error and the maximum voltage amplitude and load angle error, are better when using the neural network estimation. So, these values are 0.15% instead of 0.21%, 0.27% instead of 0.32% and 4.3% instead of 5.0%. The speed improvement is over 18,500 times, the estimation can be completed in 0.78 seconds instead of 14,632 seconds.

The mentioned results are summarized in Table I with green color indicating the WLS-DNN run pairs where the results improved. Yellow color represents unchanged results, and red color represents estimation deterioration.

TABLE I
COMPARISON OF WLS-BASED AND DNN-BASED SOLUTIONS

		18,680		44,333		44,600		20,667	
		V	θ	V	θ	V	θ	V	θ
Maximum relative error [%]	WLS	0.27	4.5	0.30	4.5	0.26	5.0	0.32	5.0
	ANN	0.14	1.2	0.24	2.8	0.26	4.3	0.27	4.3
Average relative error [%]	WLS	0.21	1.8	0.22	1.6	0.21	3.2	0.21	1.4
	ANN	0.13	0.7	0.16	1.5	0.15	2.2	0.15	1.5
Running time [s]	WLS	4620		7550		9072		14632	
	ANN	0.22		0.24		0.72		0.78	

The runtime significantly decreased in all cases, and overall estimation accuracies greatly improved. Although the sample size is small, it can be inferred that the method presented in this article yields improvements in all cases, but it is most effective for smaller networks. The future research will also focus on this aspect.

The radar diagrams of supply areas 18680, 44333, 44600 and 20667 are compared in Fig. 14.

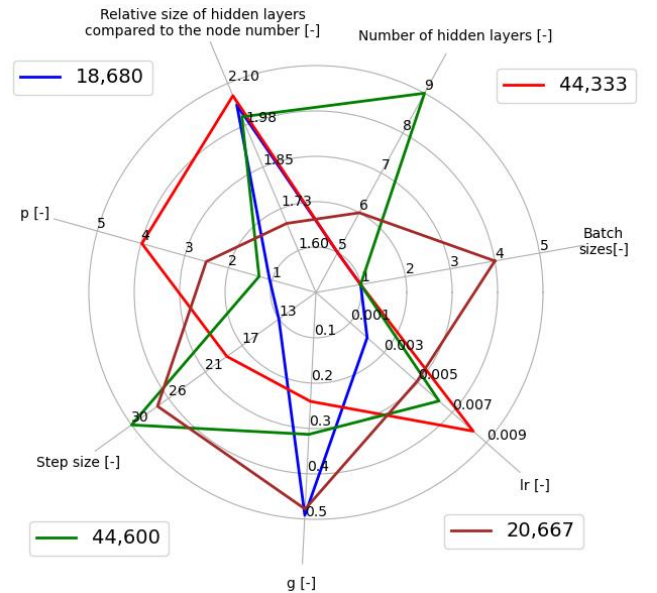


Fig. 14. Comparison of the HPO results of the supply areas.

In case of the largest transformer area, the relative size of the hidden layers compared to the node number is smaller than in

the case of the smaller areas. However, in the case of depth, the opposite characteristic can be observed in relation to network size, although the relationship is not as clear there because the deepest network belongs to the second largest area, not the largest one. Larger batch size can only be applied in the case of the most complex network, while smaller gamma typically comes with a higher learning rate or larger step size, and vice versa. And the p is greater in those cases where the tree representing the supply area with the transformer's secondary side as its root has a structure that is wider rather than deeper. However, it's worth noting that these observations are limited to these 4 LV supply areas in Western Hungary. Therefore, investigating this in other networks could yield different results.

IV. BENCHMARKING WITH OTHER DNN BASED WORKS IMPLEMENTED IN REAL ENVIRONMENT

As it was mentioned in the Introduction there are 2 comparable projects found in the literature: the evoLVDSO and the LV SCADA. According to Teixeira, Barbeiro, et al. [23], the use of traditional algorithms is not effective enough due to the lack of complete topological and electrical parameter information in their LV networks. They use 12 weeks of data, which are divided into 8-1-3 ratio for training, autoencoder training, and evaluation [24]. They work with a fixed neural architecture, using a single-layer autoencoder, and they perform the training using Evolutionary Particle Swarm Optimization [25], but they do not deal with hyperparameter optimization.

In this work, only two metering devices were used for voltage amplitude and load angle estimation per area, while in the aforementioned projects at least five or six devices were used and only voltage amplitudes were estimated. Despite this, a significantly lower average error (0.13%-0.16% instead of 0.44%-1.57%) was achieved in the comparable amplitude estimations. When the number of meters used in the cited two projects is more than the minimum, the estimation errors come to the same order of magnitude. These results were achieved with the help of TPE, which has not yet been used in this field. As described in section II/D. TPE allows hyperparameters to be optimized to be able to find the best weight parameter configuration during training. Furthermore, since HPO algorithms are effective at identifying optimal categorical parameters such as the best voltage amplitude measurement nodes, their utilization can lead to cost savings in terms of required hardware investments.

V. SUMMARY

As it was mentioned in Section I., this work, which is the extension of the project of the MTA-BME Lendület FASTER Research Group [14], is pioneering in terms of optimizing DNN hyperparameters for DSSE. Furthermore, it is important to note that the use of DNN technology for DSSE in real LV networks is still rare.

In this work, significant progress was made compared to the previous work of the authors [14]. On the one hand, the estimations of voltage amplitudes and load angles are better. The relative error of the state variables with the largest error in each area was significantly (14-73%) reduced. On the other hand, more than a four order of magnitude improvement in

running time was achieved. This improvement is due to the DNN architecture's ability to perform parallel computations with the help of GPU usage. Thus, with a dedicated GPU with sufficient memory, all runs can be performed in parallel if desired. The other reason for the speed is that the applied algorithm is non-iterative and therefore provides a straightforward solution. This speed is especially useful for example in case of renewable energy integration planning, when a large number of probability-based simulations must be made. Real-time monitoring is also becoming essential on distribution networks, and the technology presented in this paper can help this as well.

Beyond the network size mentioned in Section III., future research will continue to analyze hyperparameters for DNNs that perform DSSE. New type of layers like Batch Normalization, Convolutional or Dropout, activation functions like Leaky ReLU, Elastic Exponential Linear Unit or Mish, and architectures like Long-Short Term Memory, Gated Recurrent Unit or Graph Neural Network going to be examined with HPO. The strategy for learning rate decay and parameter optimization may be improved as well, given that stochastic HPO is a very new approach in the field of DNN-based DSSE.

REFERENCES

Examples:

- [1] A. Primadianto and C.-N. Lu, "A review on distribution system state estimation," *IEEE Trans. Power Syst.*, vol. 32, no. 5, pp. 3875–3883, Sep. 2017.
- [2] F. Ahmad, A. Rasool, E. Ozsoy, R. Sekar, A. Sabanovic, and M. Elita, "Distribution system state estimation—A step towards smart grid," *Renew. Sustain. Energy Rev.*, vol. 81, pp. 2659–2671, Jan. 2018.
- [3] M. Majdoub, J. Boukherouaa, B. Cheddadi, A. Belfqih, O. Sabri, and T. Haidi, "A review on distribution system state estimation techniques," in *2018 6th International Renewable and Sustainable Energy Conference (IRSEC)*, 2018, pp. 1–6.
- [4] K. R. Mestav, J. Luengo-Rozas, and L. Tong, "State estimation for unobservable distribution systems via deep neural networks," in *IEEE Power Energy Society General Meeting*, Portland, OR, USA, Aug. 5–9, 2018, pp. 1–5.
- [5] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 12–19, Feb. 2011.
- [6] H. Mosbah and M. E. El-Hawary, "Optimization of neural network parameters by stochastic fractal search for dynamic state estimation under communication failure," *Electr. Power Syst. Res.*, vol. 147, pp. 288–301, Jun. 2017.
- [7] R. Christie, May 1993. "IEEE 118 Bus Test Case," University of Washington, https://labs.ece.uw.edu/pstca/pf118/pg_tca118bus.htm
- [8] R. Christie, May 1993. "IEEE 14 Bus Test Case," University of Washington, http://labs.ece.uw.edu/pstca/pf14/pg_tca14bus.htm
- [9] L. Zhang, G. Wang, and G. B. Giannakis, "Real-time power system state estimation and forecasting via deep unrolled neural networks," *IEEE Trans. Signal Process.*, vol. 67, no. 15, pp. 4069–4077, Aug. 2019.
- [10] M. V. Sebastian et al., "LV state estimation and TSO--DSO cooperation tools: results of the French field tests in the evoLVDSO project", *CIREDOpen Access Proceedings Journal*, vol. 1, pp. 1883–1887, 2017.
- [11] P. Barbeiro et al., "LV SCADA project: In-field validation of a distribution state estimation tool for LV networks," *CIREDO Workshop 2016*, Helsinki, 2016, pp. 1-4.
- [12] O. Kundacina, M. Cosovic, and D. Vukobratovic, "State estimation in electric power systems leveraging graph neural networks," *arXiv preprint arXiv:2201.04056*, 2022.
- [13] D. Mukherjee, S. Chakraborty, and S. Ghosh, "Power system state forecasting using machine learning techniques," *Electrical Engineering*, pp. 1–23, 2021.
- [14] G. B. Békési, L. Barancsik, I. Táci, and B. Hartmann, "Benchmarking Various Pseudo-Measurement Data Generation Techniques in a Low-

- Voltage State Estimation Pilot Environment,” *Applied Sciences*, vol. 12, no. 6, p. 3187, Mar. 2022.
- [15] Z. Érsek, Aug. 2021. “Standard Load Profiles,” E.ON, https://www.eon.hu/content/dam/eon/eon-hungary/documents/hatarozatok-szabalyzatok-aram/eloszttoi-szabalyzat-kizarolag-honlapon-publikalando-mellekletei/Profilok_2021_teny_v1.2.xlsx
- [16] O.M.SZ, Jan. 2021, “Climate Data Series,” O.M.SZ. Hungarian Meteorological Service, https://www.met.hu/eghajlat/magyarorszag_eghajlata/eghajlati_adatsorok/
- [17] T. Huld, R. Muller, and A. Gambardella, “A new solar radiation database” for estimating PV performance in Europe and Africa,” *Solar Energy*, vol. 86, no. 6, pp. 1803–1815, Jun. 2012.
- [18] L. Huang, J. Qin, Y. Zhou, F. Zhu, L. Liu, and L. Shao, “Normalization techniques in training DNNs: Methodology, analysis, and application,” in *Proc. Comput. Vis. Pattern Recognit.*, pp. 1–20, Sep. 2020.
- [19] K. Fukushima, “Cognition: A self-organizing multilayered neural network,” *Biological Cybern.*, vol. 20, pp. 121–136, 1975.
- [20] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, *arXiv:1412.6980*
- [21] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, “Algorithms for hyperparameter optimization,” in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2011, pp. 2546–2554.
- [22] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, “Optuna: A next-generation hyperparameter optimization framework,” in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 1–10.
- [23] H. Teixeira et al., “A State Estimator for LV Networks: Results from the EvolvDSO Project,” in *CIREN Workshop 2016*, pp. 1–4, 2016.
- [24] 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.
- [25] V. Miranda and N. Fonseca, “EPSO-evolutionary particle swarm optimization, a new algorithm with applications in power systems,” *IEEE/PES Transmission and Distribution Conference and Exhibition*, pp. 745–750 vol. 2, 2002.