This article has been accepted for publication in 2022 4th Global Power, Energy and Communication Conference (GPECOM). This is the author's version which has not been fully edited and content may change prior to final publication.
Citation information: DOI: 10.1109/GPECOM55404.2022.9815784

Power System Inertia Estimation Using A Residual Neural Network Based Approach

M. Ramirez-Gonzalez, F. R. Segundo Sevilla, P. Korba
Institute of Energy Systems and Fluid Engineering
ZHAW Zurich University of Applied Sciences
Winterthur, Switzerland

Abstract—**The increasing penetration of non-synchronous generation into power grids is reducing the equivalent system inertia and leading to different frequency regulation and control challenges. Consequently, the monitoring and quantification of this inertia to implement actions that can keep it above critical levels have become a key issue for the stability of power systems. In this regard, a residual neural network (ResNet) based alternative is proposed and investigated in this paper to estimate the equivalent inertia of a sample system when synchronous generating units are displaced by converter-interfaced generators. The proposed ResNet model is trained according to the frequency of the center of inertia and the corresponding computed rates of change of frequency for a predefined time interval, where sudden generation outages and load step changes are considered under variations of total load demand and equivalent inertia reductions. The accuracy of the proposed approach is compared against the one achieved with the application of two traditional machine learning techniques, such as Support Vector Machine and Random Forest.**

Keywords— inertia estimation, convolutional neural network, residual neural network, frequency stability, converter-interfaced generation

I. Introduction

The large-scale integration of renewable energy resources through converter-interfaced generators (CIGs) is imposing new challenges considering power system frequency dynamics and stability. Since a decline in overall system inertia is undergone as synchronous machines are gradually and correspondingly displaced, an unacceptable rate of change of frequency (RoCoF) and important frequency deviations may occur after a relatively significant imbalance between system generation and load, leading to potential and uncontrolled cascading failures, and eventually a system collapse [1].

Due to the relevance of an adequate inertia to counteract and control initial changes in grid frequency, grid operators are already facing inertia-related challenges to avoid electricity service interruptions under operating scenarios with significant shares of non-synchronous energy sources. Therefore, the active quantification and monitoring of the available inertia to opportunely deploy support actions that can keep its value above critical levels has become a key issue for grid stability [2]-[4].

Although the computation of equivalent inertia constant and RoCoF could be somehow approximated from basic equations describing the inertial response of the grid to imbalances between production and demand, the lack of full required data and limited access to some information makes it a difficult task in practice. To deal with different related aspects, a number of alternative inertia estimation techniques have been recently proposed in the literature. For example, a phasor measurement unit (PMU) based method to determine the inertia of a power system according to ambient measurements and a model identification process is proposed in [5]. On the other hand, by exploiting the knowledge of network topology and the derivation of dynamic equivalents from PMUs spread on the system, a methodology to compute the equivalent inertia in large grids is proposed in [6] assuming perturbations caused by generator outages and load disconnections. In [7] authors present the development and experimentation of an inertia estimation approach based on traditional neural networks, which are trained using best-nominated inputs from PMUs measurements. By approximating the RoCoF after a disturbance from a first-order curve within a determined interval, and assuming that the related power imbalance is known, the inertia constant of an entire power system is evaluated in [8] using also PMU data. Alternatively, an inertia estimation technique based on convolutional neural networks (CNNs) and local frequency measurements is proposed in [9], where considered CNN architecture was trained using data samples collected from an equivalent generator model perturbed by a kind of excitation signals. A comprehensive review of several proposals for quantifying power system inertia can be found in [10].

Despite that power system stability assessment and control is in general becoming more challenging with the increasing incorporation of new elements to contribute to the development of cleaner electrical networks, the wide-spread deployment of several metering technologies such as PMUs opens up new opportunities to deal with a variety of issues in current operating environments from the point of view of data-driven applications [11]. In general, since more and more data are being collected, machine learning (ML) algorithms are playing key roles in different real-world domains. Specially, the significant advancements and performance achieved by deep learning (DL) methods have raised an enormous attention and have allowed them to become the main driver of many new and advanced technological developments [12], [13].

Actually, one of the most powerful and popular models in DL is Convolutional Neural Networks (CNNs). However, it is recognized that as the CNN's depth is increased, higher training and validation errors will result at some point due to vanishing and exploding gradient problems [13]. In order to alleviate this situation, residual neural networks (ResNets) were introduced [14]. ResNets are specific CNN model architectures where various forms of residual connections are added to skip a few convolutional layers, creating residual blocks that can improve convergence behavior and achieve highly accurate results in several complex tasks.

Based on the above, a ResNet based alternative to estimate the equivalent inertia of a test power system under low inertia scenarios (caused by the integration of CIG and displacement of synchronous generators) is presented in this work. Following a supervised learning approach, model input data and expected output were collected from time domain simulations of the power system, where a wide range of operating conditions resulting from variations in total load demand, system inertia reductions, single outages of synchronous generators and individual load step changes were taken into account. In this way, the frequency of the center of inertia and involved RoCoFs after a given disturbance were determined and used for the model training task. The effectiveness and accuracy of the proposed approach is evaluated according to the value of the coefficient of determination achieved for the test dataset, and the results are compared to the ones provided by traditional Support Vector Machine (SVM) and Random Forest (RF) based possibilities. As far as the authors know, there are no earlier studies on the use of a residual neural network framework for power system inertia approximation.

II. POWER SYSTEM INERTIA

In general, inertia is critical for maintaining the stability, reliability, and security of power systems. In this way, the more inertia the system has, the better the capability of the grid to resist dynamic changes [1]. Basically, the inertial response has to do with the immediate power system reaction to frequency deviations. It depends on the rotational energy stored in the rotors of the synchronous machines and affects the initial RoCoF following sudden power imbalances.

For representative purposes, the inertial response of an interconnected power system with a fully synchronous generation mix can be estimated from the following equation:

$$\frac{2HS_B\omega}{\omega_0^2}\frac{d\omega}{dt} = P_{\text{gen}} - P_{\text{load}} \tag{1}$$

where ω and ω_0 are the actual and nominal angular speeds, respectively, and S_B, H, P_{gen}, and P_{load}, denote correspondingly total amounts of power rating, inertia constant, generated power, and consumed electrical power. By assuming that $\omega \approx \omega_0$, equation (1) can be further simplified to the next expression:

$$\frac{2HS_B}{\omega_0}\frac{d\omega}{dt} = P_{\text{gen}} - P_{\text{load}} \tag{2}$$

Additionally, considering that $\omega = 2\pi f$, the initial RoCoF just after any power imbalance can be derived from (2), and thus approximated according to:

$$\frac{df}{dt} = \frac{P_{\text{gen}} - P_{\text{load}}}{S_B}\frac{f_0}{2H} \tag{3}$$

Based on equation (3), it can be noted for example that, after a sudden generator disconnection or load tripping, the initial RoCoF will be influenced significantly by a large difference between P_{gen} and P_{load}. Moreover, the RoCoF can be considerably high if the aforementioned incidents take place in particular regions with a relatively small inertia constant. In this regard, by grouping all involved synchronous generators into an equivalent rotating mass, the related equivalent inertia constant can be computed as:

$$H = \frac{\sum_i^G H_i \cdot S_{ni}}{S_B} \tag{4}$$

where G is the number of synchronous generators, and H_i and S_{ni} are respectively the inertia constant and rated power (MVA) of a given generator i.

III. CONVOLUTIONAL NEURAL NETWORKS (CNNS)

Convolutional Neural Networks are a type of feed-forward networks that process information arranged in a grid-like topology. They can be used not only for classification purposes but also to perform continuous data prediction. Typically, CNNs are composed of convolutional layers, pooling layers, and fully connected layers. By stacking convolution and pooling operations, meaningful features maps can be extracted from the input data, which are then fed to a set of fully connected layers to achieve the considered classification or regression tasks. This process can be simply illustrated through the diagram in Fig. 1.

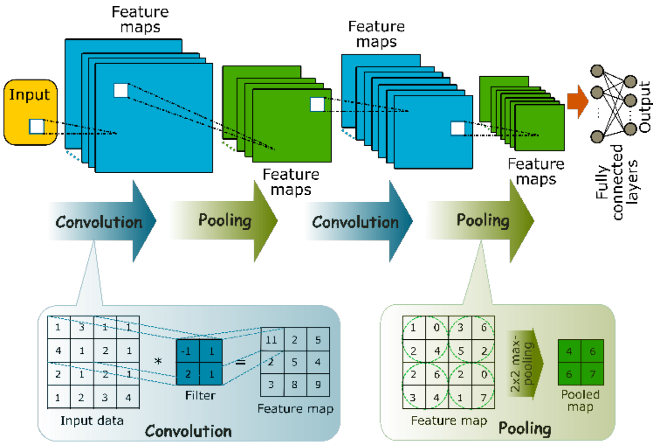

Fig. 1. Conceptual architecture of CNNs.

Convolution operations, as indicated in Fig. 1, rely on the application of simple multiplications and additions through convolutional filters. The values of these filters are automatically adjusted during network training to extract useful information. Usually, supplementary functions of the Rectified Linear Unit (ReLU) type are included after the filtering step in order to only detect and select particular features for postprocessing. On the other side, the pooling operations are non-parametric. They are used mainly to reduce the spatial dimensions of the received data by for example taking the biggest value within a region according to a predefined window. Some other operations such as Batch Normalization (BN) and Dropout are typically added to the ones in Fig. 1 to provide better performance. BN may allow the use of relatively higher learning rates and therefore improve the convergence speed of the network. Dropout enhances model generalization ability and prevents overfitting [12].

IV. RESIDUAL NEURAL NETWORKS (RESNETS)

Basically, residual neural networks refer to a class of CNNs where skip connections are incorporated to skip one or more convolutional layers [14]. In general, although a large number of hidden layers may facilitate the modeling of very complex underlaying representations (because of the total number of parameters), lower training accuracies may be attained at some point as the deep of the network is increased. This phenomenon has to do with a vanishing/exploding gradient problem. During the training phase, the gradient might be close to cero or very large when information is transmitted layer by layer from the output to the input of the network.

A. *ResNet basic principle*

Residual neural networks introduce residual modules to allow the flow of information to different layers without attenuation, influencing the effective depth of the network for training purposes and alleviating the accuracy degradation of deep networks. A basic residual module can be represented as in Fig. 2.

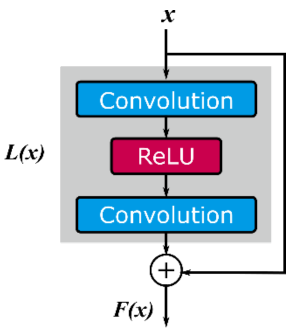

Fig. 2. Basic residual module.

According to Fig. 2, a given input x is used to obtain the output $L(x)$ based on two convolutional blocks here, and then both x and $L(x)$ are added to compute $F(x)$, which denotes the output of the residual module. The idea behind this is that instead of expecting that some stacked layers approximate a desired undelaying mapping $F(x)$, they are reformulated to fit a related residual mapping by incorporating shortcut connections that may skip one or more layers. In fact, these connections perform identity mappings, and ideally a given residual mapping will be pushed to zero if the identity mapping turns out to be optimal, which is easier than learning the identity function with a stack of convolutional blocks [14], [15]. By looking at the configuration in Fig. 2, the skip connection will make the surrounded network blocks to learn the residual function $L(x)$ expressed by:

$$L(x) = F(x) - x \tag{5}$$

In general, skip connections facilitate the development of deep neural models since training is allowed to go directly to some other layers. It is clear that for the computation of $F(x)$ in Fig. 2, the shape of x and $L(x)$ must match. Therefore, if that is not the case, x has to be transformed as required. A residual network architecture can be composed of several or many residual modules.

B. *Improved residual model*

In order to enhance the training and the generalization capability of ResNets, various paths for the propagation of information through the entire network and different arrangements for the residual module were investigated in [16]. Then, after extensive experimentation and analysis, the design of the residual configuration illustrated in Fig. 3 was derived.

It is seen form Fig. 3 that the residual module in this case comprises a series of Batch Normalization, ReLU, and Convolutional blocks, where the input to the next residual module in the network will be the output of the Addition block x_{l+1}, where l refers to the l-th stacked module. According to [16], the use of the residual configuration in Fig. 3 leads to an improved optimization process and better regularization of network models.

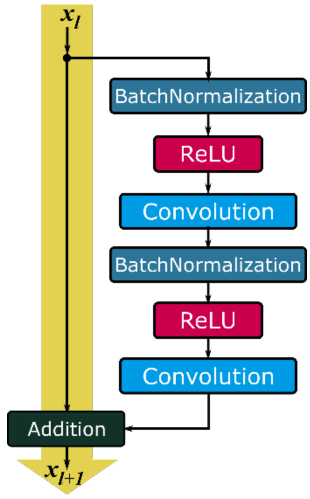

Fig. 3. Full pre-activation arrangement.

V. ResNet Based Model For Inertia Estimation

A. *Dataset for the studies*

Since data provide the main source of learning in any ML algorithm, a dataset needs to be available to build a ML model for a given prediction task. In this case, and for illustrative purposes, the New England Power System shown in Fig. 4 is considered for the dataset generation. This is a benchmark system that represents a simplified version of the high voltage transmission grid in the northeast of the U.S.A. It consists mainly of 39 buses, 10 synchronous generators, 19 loads, 34 transmission lines and 12 transformers. For the simulation studies here, all system parameters were taken from [17].

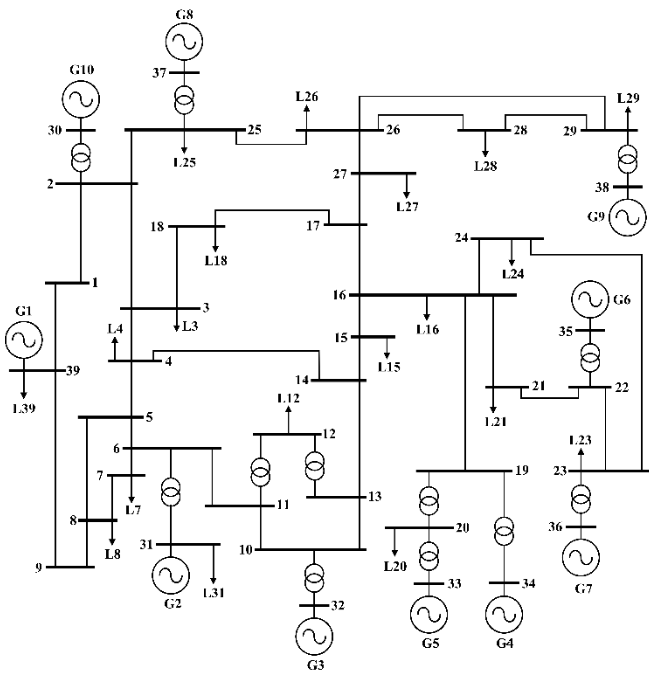

Fig. 4. Sample power system.

A supervised learning approach is followed in this work for model development. In this regard, the input data to the model and the expected output were collected as described next. From the initial base case scenario available for the studies, new operating conditions were simulated according to changes in total system load demand and generation. For this purpose, the active power of all the loads was varied between

50% and 100% of the base case value, and the active power of generators was randomly altered within a corresponding range using a normal distribution. All these changes were considered under scenarios of system inertia reduction from 100% to 50%, where the inertia of each generator was accordingly modified, and the equivalent system inertia was computed from expression (4) in Section 2.

For every system load level involved and every inertia reduction scenario, the single outage of generators and the step increase of loads (one at a time) were considered as sudden power imbalances. In this manner, an average frequency of the system was determined after the disturbance based on the following equation, where f_{COI} represents the frequency of the center of inertia [18] of the grid under study:

$$f_{\mathrm{COI}} = \frac{\sum_{i=1}^{G} H_i \cdot S_{ni} \cdot f_i}{S_B} \tag{6}$$

It should be noted that the computation of f_{COI} during the simulated events was carried out based on the generators still connected to the grid. It is also important to mention that, for the generation of the training dataset, no CIG units were incorporated into the system at this stage since the reduced inertia scenarios were accomplished by directly varying the inertia constant of the included synchronous generators.

In this study, f_{COI} given by (6) was recorded every 20 ms for a time period of 2.5 s after the disturbance, and the consequent RoCoF was computed and collected at each sampling step. An illustration of the type of information obtained from a generator outage event for variations in the equivalent inertia of the sample system is shown in Fig. 5.

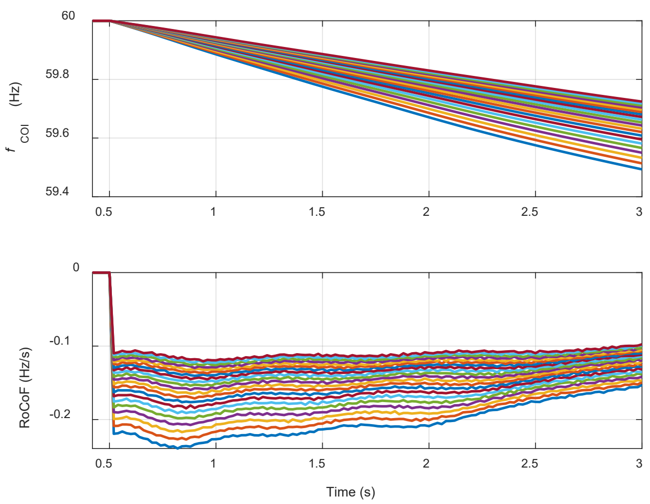

Fig. 5. System response after generator outage and *H* variations.

Both f_{COI} and RoCoF were considered together here to be used as inputs to the ML model described in the next subsection. For model training purposes, the gathered observations were associated with corresponding target outputs, which in this study refer to the equivalent inertia constant determined from (4). For the purpose of the work presented here, 2951 input examples were collected and included in the dataset.

B. Model structure

The ResNet model considered in the studies is illustrated in Fig. 6, which comprises two residual modules according to the arrangement in Fig. 3. In this case, all the convolutional blocks in Fig. 6, referred as Conv2D, were configured with 64 filters. Now, while the kernel size for the first of these blocks is 2x4, a 1x4 size is used in the rest of them (a stride of one is applied in any case). Moreover, ReLU functions were incorporated here into the Activation blocks shown in red. On the other hand, a max-pooling operation identified as MaxPooling2D, with a kernel size of 1x4 and a stride of one, was employed just before and also after the two stacked residual modules. In addition, dropout regularization, with a dropout rate of 0.3, was inserted before flattening the pooled feature maps obtained from the second MaxPooling2D block. Finally, one dense layer composed of 100 neurons with ReLU activation, and one neuron output layer with a linear activation function was considered in the last processing phase.

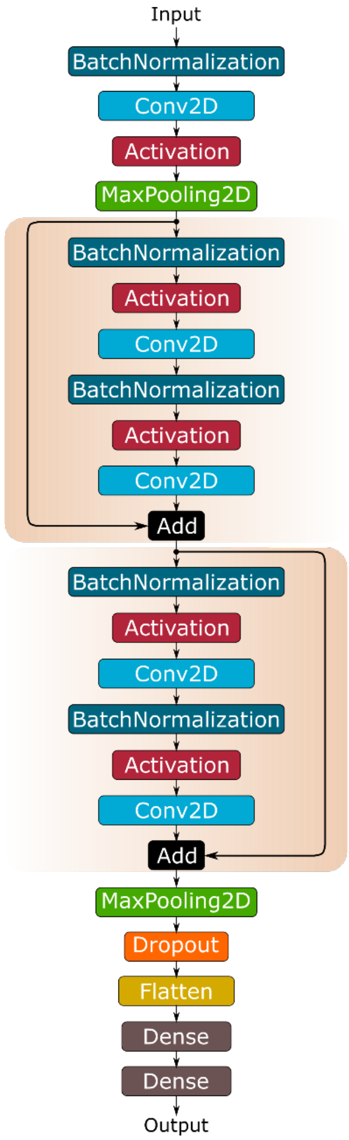

Fig. 6. Considered model structure.

All the BatchNormalization blocks in Fig. 6 contribute to stabilizing and boosting the learning process. In addition, it is worth mentioning that, for the studies presented here, the parameters of the considered model were selected based on author's previous experience. However, it is recognized that a hyperparameter optimization approach might also be applied in this regard as an option for an enhanced and systematic design.

VI. Simulation Results

A. Model training and validation

First of all, the representative input examples of the collected dataset were rescaled in the range 0-1 prior to be used in the proposed ResNet model. After that, the available samples were randomly split into training, validation, and test sets, with a corresponding ratio split of 50/20/30. Therefore, 50% of the data were selected to search for the optimal weights of the model, 20% for model evaluation during training, and 30% for testing purposes after training completion. In order to guide the learning process, the loss function known as Mean Squared Error was employed to assess the error between model outputs and targets. Moreover, the Adam optimizer was selected for loss function minimization [19]. In addition, the number of epochs to train the model was set to 250 with a batch size of 10.

After several runs with a random initialization of the network's weights, Fig. 7 illustrates the best learning performance achieved by the proposed approach during the training phase. It can be observed that the validation curve closely tracks the training curve, with a relatively fast and stable convergence to a minimum value as the epochs increase, which can be identified as a relatively good fit.

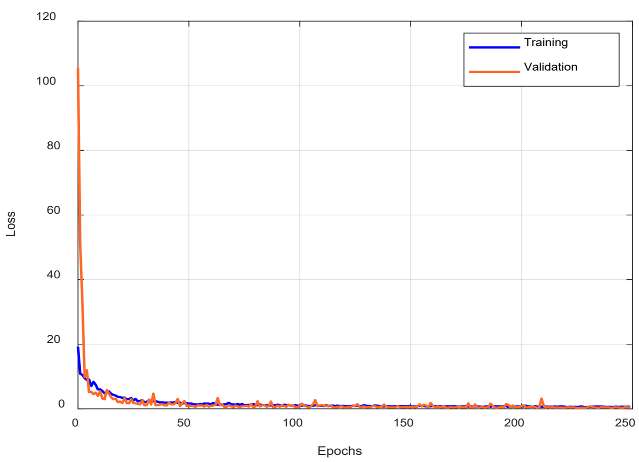

Fig. 7. Loss curves.

B. Model testing and evaluation

The effectiveness of a machine learning model depends on its generalization ability and performance when applied to new and unseen information. In this sense, such model may fit the training data with a high level of accuracy but provide poor results when new data samples are presented. In this study, the model's ability to generalize to new cases is evaluated with the reserved test dataset aforementioned in the previous subsection.

Fig. 8 portrays the performance of the model in this case, where the results of predicted versus true inertia values are plotted along with the reference line for perfect predictions. In order to quantitatively evaluate the accuracy of the obtained network's outputs, the coefficient of determination R^2 [20] was computed according to the following expression:

$$R^2 = 1 - \frac{\sum_{i=1}^{n_{samples}} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n_{samples}} (y_i - \bar{y})^2} \tag{7}$$

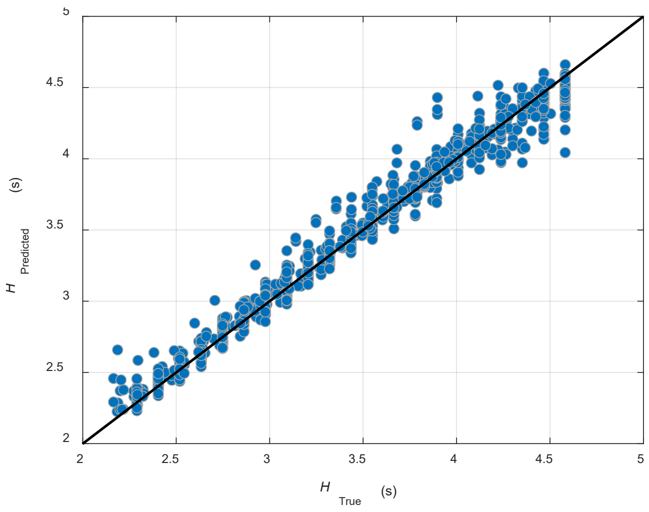

Fig. 8. Predicted vs. actual test inertia values.

where y_i represents the target value, $\hat{y}_i$ refers to the model prediction, and $\bar{y}$ indicates the mean value of the desired outputs. In this way, a value of 0.978 was achieved for the test set here (a score of 1.0 denotes a perfect fit).

Now, by looking at the corresponding error histogram in Fig. 9, it can be observed that most of the estimation errors fall within ±5%, and that there are only a very few instances where the related absolute error is greater than 10%.

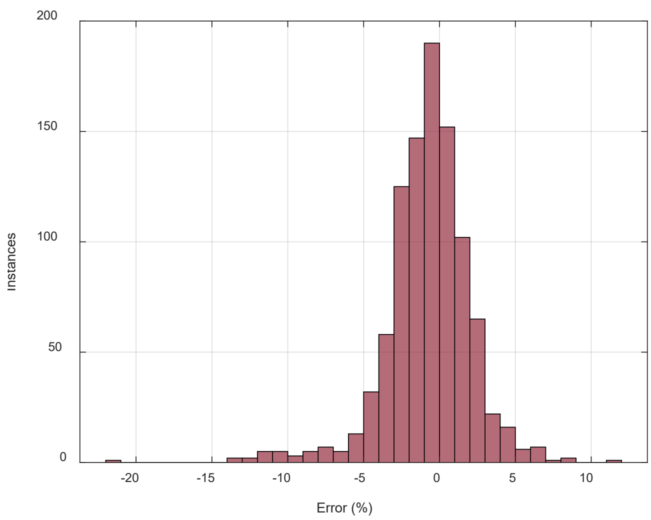

Fig. 9. Error histogram.

In terms of the R^2 measure, the results achieved with the proposed approach were compared against the ones obtained with the application of two traditional machine learning techniques, such as Support Vector Machine (SVM) and Random Forest (RF) [21]. SVM is a popular ML algorithm based on kernels that allow to transform a low-dimensional input space into a high-dimensional one, as a tactic to convert a non-separable problem into a separable one. On the other side, RF is based on ensemble learning, and basically uses a collection of decision trees to produce a solution to a given regression or classification problem. In this regard, Table I shows the obtained results with the considered approaches, where it can be noted that the proposed ResNet based alternative is able to provide a better performance for the task addressed in this work.

TABLE I. R^2 RESULTS FOR THE TEST SET

Model	R^2
Proposed	0.978
SVM	0.950
Random Forest	0.907

It is important to remember at this point that, for the collection of the training data set, the inertia constants of the synchronous machines were modified as a mean to reduce the equivalent inertia of the system, but none of these generators were replaced by CIG units. However, for further evaluation of the trained ResNet model, simulations were carried out considering the displacement of these components with solar PV generators. In this sense, Table II provides the obtained results for some illustrative cases, where the integration level of solar PV generation (SPVG), the number of original synchronous units being replaced, the equivalent inertia values, and the associated errors are included.

TABLE II. PREDICTION RESULST WITH CIG INTEGRATION

SPVG (%)	Number of displaced units	*H* (s)		|Error| (%)
		Actual	Predicted	
19.70	2	4.13	4.07	1.45
29.99	3	3.97	3.98	0.25
39.94	4	3.73	3.90	4.55
50.49	5	3.60	3.77	4.72

The share of SPVG in Table II refers to the percentage of overall generation power being accommodated by SPVG through the direct substitution of synchronous generators. In any case, only remaining and connected synchronous units were used for the computation of f_{COI} and the RoCoF, which were used later as inputs for model prediction. Based on the approximations of the proposed model, it can be observed from Table II that an absolute error of less than 5% was achieved in each of these sample cases.

VII. CONCLUSIONS

On the basis of residual neural network concepts, a data-driven approach for the estimation of power system equivalent inertia has been presented in this study. The proposed model was trained in a supervised manner, with a set of input examples composed of the f_{COI} and the RoCoFs observed after considered disturbances. A wide range of operating conditions resulting from variations in total load demand and system inertia reductions were used for this purpose. To facilitate the generation of representative model inputs, low inertia scenarios were simulated by directly changing the inertia of each synchronous generator in the system in a proportional way. However, the actual substitution of these generators by SPVG units was in fact carried out as part of the model evaluation stage.

By using the proportion of the dataset included in the test split and assessing the model performance in terms of the coefficient of determination, the proposed approach was able to provide relatively better results than two other possibilities based on SVM and RF techniques. Besides, under scenarios of true integration of SPVG and actual displacement of synchronous generators, the proposed alternative achieved absolute prediction errors of less than 5% (for the illustrative cases considered here), which shows its potential application. The obtained results in this work are intended to be used in further research considering a lab-scale experimental setup.

REFERENCES

[1] J. Machowski, Z. Lubosny, J. W. Bialek, J. R. Bumby, Power System Dynamics Stability and Control. NJ: Wiley, 2020

[2] A. Dixon, Modern Aspects of Power System Frequency Stability and Control. UK: Academic Press, 2019.

[3] A. Monti, F. Milano, E. Bompard, X. Guillaud, Converter-Based Dynamics and Control of Modern Power Systems. UK: Academic Press, 2021.

[4] P. Denholm, T. Mai, R. Wallace, B. Kroposki, M. O'Malley. "Inertia and the Power Grid: A Guide Without the Spin", Tech. Rep. NREL/TP-6A20-73856, 2020.

[5] K. Tuttelberg, J. Kilter, D. Wilson, K. Uhlen, "Estimation of Power System Inertia From Ambient Wide Area Measurements," IEEE Transactions on Powewr Systems, vol. 33 (6), pp. 7249-7257, 2018.

[6] G. Rossetto, F. Pozzi, V. Ilea, A. Berizzi, E. Carlini, G. Giannuzzi, R. Zaottini, "Measurement-based inertia estimation method considering system reduction strategies and dynamic equivalents," IEEE Milan PowerTech, pp. 1-6, June 2019.

[7] E. Raju, H. Marzooghi, J. Yu, V. Terzija, "Development and Validation of Artificial Neural Network-Based Tools for Forecasting of Power System Inertia With Wind Farms Penetration," IEEE Systems Journal, vol. 14 (4), pp. 4978-4989, 2020.

[8] M. Hayerikhiyavi, A. Dimitrovski, "A practical assessment of the power grid inertia constant using PMUs," 52nd North American Power Symposium, pp. 1-5, April 2021.

[9] A. Poudyal, R. Fourney, R. Tonkoski, T. Hansen, U. Tamrakar, R. Trevizan, "Convolutional Neural Network-based Inertia Estimation using Local Frequency Measurements," 52nd North American Power Symposium, pp. 1-6, April 2021.

[10] B. Tan, J. Zhao, M. Netto, V. Krishnan,V. Terzija, Y. Zhang, "Power system inertia estimation: Review of methods and the impacts of converter-interfaced generations," International Journal of Electrical Power and Energy Systems, vol. 134 (4), 2021.

[11] Y. Xu, Y. Zhang, Z. Yang, R. Zhang, Intelligent Systems for Stability Assessment and Control of Smart Power Grids. FL: CRC Press, 2019.

[12] H. Osupyan, B. I. Edwards, A. D. Cheok, Deep Neural Network Applications. FL: CRC Press, 2022.

[13] M. Sewak, M. Rezaul, P. Pujari, Practical Convolutional Neural Networks. UK: Packt Publishing, 2018.

[14] K. He, X. Zhang, S. Ren, J. Sun, "Deep Residual Learning for Image Recognition," IEEE Conference on Computer Vision and Pattern Recognition, pp. 1-9, June 2016.

[15] I. Zafar, G. Tzanidou, R. Burton, N. Patel, L. Araujo, Hands-On Convolutional Neural Networks with TensorFlow. UK: Packt Publishing, 2018.

[16] K. He, X.Zhang, S. Ren, J. Sun, "Identity Mappings in Deep Residual Networks," European Conference on Computer Vision, pp. 630-645, Sept. 2016.

[17] DIgSILENT PowerFactory2020, Technical Reference Documentation - 39 Bus New England System, Gomaringen, Germany.

[18] S. You, H. Li, S. Liu, K. Sun, W. Wang, W. Qiu, Y. Liu, "Calculate Center-of-Inertia Frequency and System RoCoF Using PMU Data," IEEE Power & Energy Society General Meeting, pp. 1-5, July 2021.

[19] D. P. Kingma, J. L. Ba, "Adam: a method for stochastic optimization," in Proc. 3rd International Conference for Learning Representations, pp. 1-15, July 2015.

[20] D. Sarkar, R. Bali, T. Sharma, Practical Machine Learning with Python: A Problem-Solver's Guide to Building real-World Intelligent Systems. NY: Apress, 2018.

[21] A. C. Müller, and S. Guido, Introduction to machine learning with Python – A guide for data scientists. CA: O'Reilly Media, 2017.