Learning a Generator Model from Terminal Bus Data

Nikolay Stulov*, Dejan J Sobajic†, Yury Maximov‡*, Deepjyoti Deka‡, and Michael Chertkov‡*
*Skolkovo Institute of Science and Technology, Moscow, Russia
† Grid Consulting LLC, San Jose, CA, USA
‡ Theoretical Division and Center for Nonlinear Studies, Los Alamos National Laboratory, Los Alamos, NM, USA

arXiv:1901.00781v1 [stat.ML] 3 Jan 2019

***Abstract*—In this work we investigate approaches to reconstruct generator models from measurements available at the generator terminal bus using machine learning (ML) techniques. The goal is to develop an emulator which is trained online and is capable of fast predictive computations. The training is illustrated on synthetic data generated based on available open-source dynamical generator model. Two ML techniques were developed and tested: (a) standard vector auto-regressive (VAR) model; and (b) novel customized long short-term memory (LSTM) deep learning model. Trade-offs in reconstruction ability between computationally light but linear AR model and powerful but computationally demanding LSTM model are established and analyzed.**

I. Introduction

Power Generator (PG) is a complex device that delivers energy to power grids. PG is described by a system of non-linear differential-algebraic equations (DAEs) of sufficiently high order, see e.g. [1]. The generator is connected to the rest of the power system (PS) through the terminal bus.

In this manuscript we assume that active and reactive powers (P and Q, respectively) as well as voltage V and phase φ are observed at the terminal bus with a sufficient frequency. We then learn the generator model as a mapping between a pair of input variables (P, Q) and a pair of output variables (φ, V). We do not assume any prior knowledge of the generator power-engineering details and thus experiment with two generic ML approaches developed for correlated linear and nonlinear time-series in an application-agnostic context.

Our "PG-agnostic" approach should be contrasted with "PG-informed" studies, some dated back to the 1980s [2], [3], [4], [5], reconstructing parameters in the corresponding system of DAEs. Complexity of the DAEs makes the reconstruction task difficult and thus computationally expensive. In many on-line power engineering applications, where speed of prediction is primary to quality and interpretability, the PG-agnostic light and computationally efficient solutions may provide a desired compromise for the reconstruction task.

To accomplish the task we have built two schemes: vector auto-regressive (VAR) scheme, which is fast and simple, yet limited due to underlying linearity assumption; and a recurrent neural network (RNN) scheme of the long short-term memory (LSTM) type, which is superior in the reconstruction quality, yet more expensive implementation-wise.

The rest of the manuscript is organized as follows. Technical introduction, with primers on AR, LSTM and data generating procedure, is given in Section II. Section III describes in sufficient detail our VAR and LSTM algorithmic solutions. Finally, data generation and learning experiments are described and analyzed in Section IV.

II. Technical Introduction

In this section we revisit classical algorithms for time series prediction and review our custom NN LSTM solutions for the problem.

A. Auto-regressive process

Auto-regressive (AR) model is a proven tool in many stationary time series applications. In its multivariate version AR model becomes vector auto-regressive model (VAR) [6] described by the following equation

$$\mathbf{y}_t = \mu + \sum_{i=1}^{p} \mathbf{A}_i \mathbf{y}_{t-i} + \varepsilon_t, \tag{1}$$

where $\mathbf{y}_t$ is the time series, μ and $\mathbf{A}_i$ are the parameters to be learned, ε_t is a stochastic white noise process and p is the order of the VAR model.

When the series depend on some other exogenous, changing in time characteristics, $\mathbf{x}_t$, the model becomes

$$\mathbf{y}_t = \mu + \sum_{i=1}^{p} \mathbf{A}_i \mathbf{y}_{t-i} + \beta \sum_{i=1}^{p} \mathbf{A}_i \mathbf{x}_{t-i} + \varepsilon_t, \tag{2}$$

where β is an additional parameter which needs to be learned.

B. Long Short-Term Memory Network

Fully-connected neural networks (NNs) are powerful state-of-the-art tools for many ML tasks. In order to improve NNs' task-specific generalization ability new architectures are proposed. For example, convolutional neural networks (CNNs) were designed for images [7]. CNNs take advantage of the image (postulated) translational invariance, thus resulting in a significant reduction in the number of training parameters. In the case of a time-stationary data sequence similar reduction in the number of training parameters is achieved through the so-called Recurrent Neural Networks (RNNs) proposed in [8], characterized by time-invariant (or quasi-time invariant, i.e. changing slowly with time) couplings between the signal at different time-frames. However, earlier tests of RNN have also revealed a number of significant drawbacks in performance. The long short-term memory (LSTM) [9] NN extending RNNs

with the concept of memory cell were designed to overcome such issues. Memory cell enables RNN to learn when to remember and when to forget previously read data. As a result, LSTMs offer better performance in the task of modelling long time series.

C. Synthetic Data for the Ground Truth

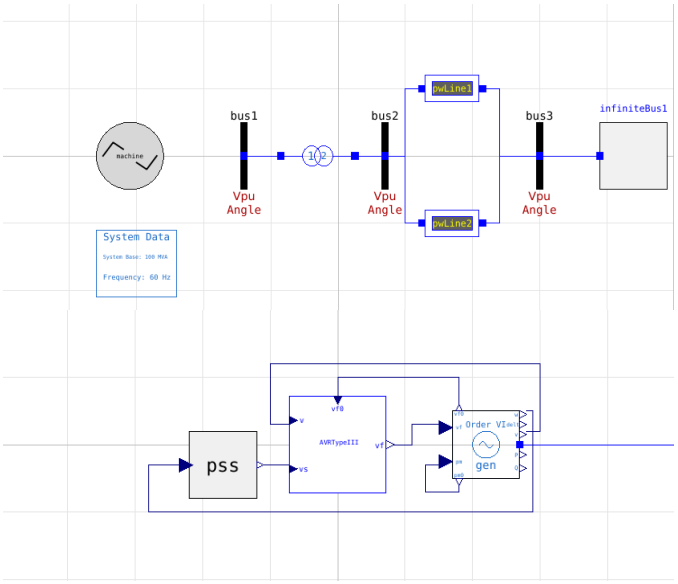
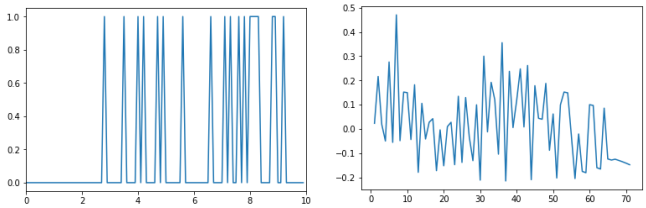


Fig. 1. Example grid (top) and machine (bottom) from [1] (example 13.2).

In this work we choose to work with synthetically generated data. The data was generated for a typical power generator with OpenModelica using openly distributed package OpenIPSL [10]. Specifically, we use OpenIPSL implementation of a generator in a simple grid example 13.2 from [1], illustrated in Fig. 1. The model is initiated with static power flow (PF) values calculated with PSAT [11]. To imitate uncertainty, exogenous to the generator, we add stationary perturbations to the model in the form of stochastic faults described in the following. [1] The faults are introduced at the connection between the second bus and the first power line in Fig. 1. In our experiments we set fault resistance to 0.01 pu and select fault reactance such that the generator survives the fault (does not disconnect), while showing some visible non-linearity. We later alter these parameters to test our models in different regimes (linear and nonlinear) and fine-tune their performance and robustness. We find out, that resistance value of 10^{-5} is the smallest which can be stabilized by the generator (also equipped with the Power System Stabilizer). We also alter the parameters of the generator randomly by sampling values from the normal distribution, $\mathcal{N}(v, 0.1v)$, where v is the static value used in the previous experiments.

Dynamic stochasticity is introduced into the model by scheduling faults according to the following pair of master

[1]Notice that this choice of uncertainty is due to the fact that other characteristics of the power grid are hard-coded in the version of OpenModelica and OpenIPSL available to us, not allowing to inject into the model endogenously varying characteristics in any other way then generating a sequence of faults.



(a) Process (b) Autocorrelation

Fig. 2. Telegraph process.

equations describing statistics of a stochastic stationary telegraph process

$$\begin{cases} \frac{\partial}{\partial t}\mathbb{P}(a,t\,|\,x,t_0) &= -\lambda\mathbb{P}(a,t\,|\,x,t_0) + \mu\mathbb{P}(b,t\,|\,x,t_0) \\ \frac{\partial}{\partial t}\mathbb{P}(b,t\,|\,x,t_0) &= \lambda\mathbb{P}(a,t\,|\,x,t_0) - \mu\mathbb{P}(b,t\,|\,x,t_0). \end{cases} \tag{3}$$

Note that solution of Eq. (3) initialized with, $P(a,0) = 0$ and $P(b,0) = 1$ (no fault at $t = 0$), becomes

$$\begin{cases} \mathbb{P}(a,t\,|\,\pi) &= \frac{\mu}{\lambda+\mu} + \frac{\lambda\pi_a - \mu\pi_b}{\lambda+\mu}e^{-(\lambda+\mu)t} \\ \mathbb{P}(b,t\,|\,\pi) &= \frac{\lambda}{\lambda+\mu} + \frac{-\lambda\pi_a + \mu\pi_b}{\lambda+\mu}e^{-(\lambda+\mu)t}, \end{cases} \tag{4}$$

where the first and second terms in the sum correspond, respectively, to the stationary values, and to the exponentially decreasing with time finite memory correction. Eqs. (3) are included (coded) explicitly in the computational scheme so that at each time step, the fault state is sampled from the resulting distribution. It is straightforward to check that the result of this scheme is the generation of a Bernoulli distributed random variable with success probability $\mathbb{P}(a,t\,|\,\pi)$. In addition we also randomize at each step the resistance and reactance of the fault.

We set $\mu = 0.03$ and $\lambda = 0.02$ to bias towards an open fault and to randomize the time of the first fault occurrence. An examplary process and its auto-correlation (as a function of the time lag) are shown in Fig. 2a and Fig. 2b respectively. The auto-correlation is computed as Pearson correlation, Eq. (5), between x at the observation time and x at the time shifted by a lag. Observe that the auto-correlation resembles white noise, which indicates that the process is almost memoryless.

$$r_{xy} = \frac{\sum_{i=1}^{T}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{T}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{T}(y_i - \overline{y})^2}} \tag{5}$$

The OpenIPSL solver requires time resolution to be at least 10^5 steps per second. On the other hand standard PMU measurements are recorded $10 - 50$ times per second. These considerations motivate us to analyze the the practical case of 10 measurements per second.

III. Algorithms, Diagnostic and Performance

A. Vector AR

In order to determine the optimal order of VAR model we carry out several experiments computing correlations between input and output variables. For each pair of vector-variables

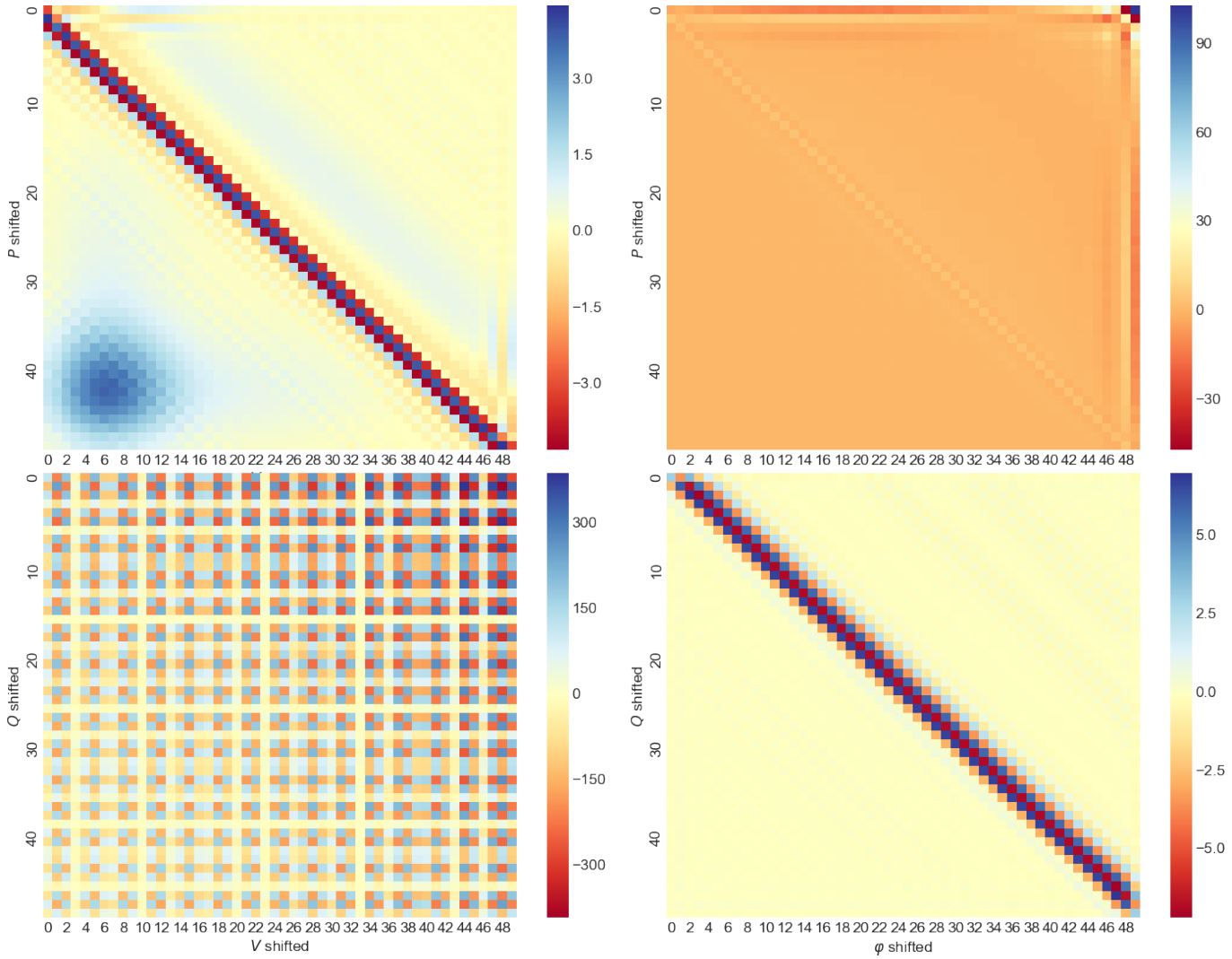


Fig. 3. Colormap interpretation of inverted correlation matrices for each pair of variables. Upper triangular part of each plot corresponds to degree of conditional dependence of first variable's present and second variable's past. Largest sequence of non-zero elements ending with zeros constitutes a measure for model order p.

(two time-series) we construct a matrix of correlations, C, between the two variables shifted by different time lags (columns for lags in first variable and rows for lags in the second). Next we invert C and investigate non-zero elements. For Gaussian noise, non-zero elements of the upper triangle of C^{-1} signifies conditional independence (lack of correlations) between first variable's present and second variable's past, while non-zero elements of the lower triangle denotes that the present value of the second variable is conditionally independent of the first variable's past [12]. The resulting C^{-1} matrices for each pair of variables are shown in the form of a heat-map in Fig. 3.

We derive from the heat maps in Fig. 3, that for all pairs of variables, but the active power P and phase φ, the length of strip of non-zero elements does not exceed 5 time steps. This observation means that after 5 time steps all variables except P and φ become conditionally independent. For the case of P and φ we observe notable dependence with lags of up to 32. We conclude that this pair of variables is conditionally independent after 32 time steps. Based on this derivation, we set the order of VAR model to be equal to 32.

This decision is further supported by implementation-specific information criteria-based order selection methods. We have experimented with multiple criterion, including Akaike information criterion (AIC, see Eq. 6) [13], final prediction error (FPE) [13], Hannan-Quinn information criteria (HQIC) [14] and Bayesian information criterion (BIC) [14]. We find, that AIC yields the most stable results for our case.

$$\mathrm{AIC} = 2k - 2\ln\mathcal{L} \tag{6}$$

where k is the number of parameters and $\mathcal{L}$ is the maximum likelihood achieved by the model. It reports the same 32 time steps lag as what we have concluded from Fig. 4

Notice that the input variables, e.g. voltage and phase, are per se not statistically stationary. We observe, however, that

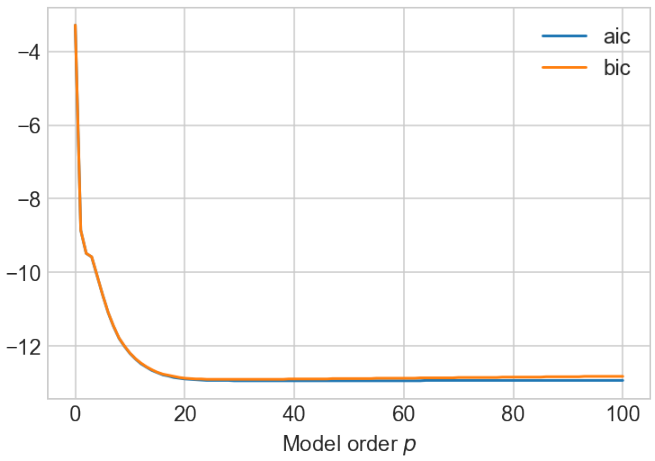


Fig. 4. Comparison of AIC and BIC.

the assumption of statistical stationarity applies much better to the characteristics' temporal derivatives (or equivalently difference/increment between the values in two adjacent time slots). Therefore, we apply VAR to the temporal increments. The LSTM model, in contrast, does not require statistical stationarity, which explains why we apply it directly to the input variables.

B. Weight-Dropped (WD) LSTM

To boost performance on the low-resolution data, we follow the state of the art in the LSTM modeling [15] and use the Weight-Dropped LSTM. Two other specifics of the scheme we use are related to the use of the multiple LSTM approach and its applications to the supervised setting (input-output relation). The multiple LSTM architecture, utilizing a number of different training techniques, e.g. dropout, hidden-to-hidden dropout, weight decay and varied sequence length, is considered empirically superior to other ML options because of its ability to capture wide and varying ranges of data dependencies. We use [16] for implementation of the supervised scheme.

IV. Experiments and Results

A. VAR model experimental setup

Based on the preliminary analysis of the data, we train a vector autoregressive model of order 32 with the input vector (P, Q) and the output vector of the (φ, V) time increments.

Generation of the input-output training data is described in Section II-C.

Since the VAR model is purely linear, it is important to investigate its dependence on the change in the noise magnitude. In order to do that, we construct 4 pairs of series with fault reactances set to zero and fault resistances set to, 10 pu, 1 pu, 0.1 pu and 0.01 pu respectively. We then train 4 models on first series of each pair and test them on the second series of the corresponding pair.

B. LSTM model experimental setup

We train the neural network to read (V, φ) pairs sequentially and output (P, Q) in a (varied-length) window. The objective functional is the mean squared error (MSE) between model outputs at each step and the ground truth data.

We train the model on 200 samples with fault parameters as described above. Then one evaluates the model on 200 samples with different and additionally randomized fault parameters. Finally, we perform further testing with randomized generator parameters and extreme noise, as discussed later.

C. Estimation of quality

To measure the algorithm quality we use normalized root mean squared error (NRMSE), which allows us to treat error as the percentage described by

$$NRMSE(x, y) = \frac{\sqrt{\frac{1}{T}\sum_{t=0}^{T} ||x_t - y_t||_2}}{\sqrt{\frac{1}{T}\sum_{t=0}^{T} ||x_i||_2}}, \tag{7}$$

where x is the ground truth, y is the estimation and $||\cdot||_2$ is the notation for the ℓ_2 norm.

D. Results

Quantitative comparison of all the models tested is summarized in Table I.

1) VAR: We begin with training and testing this model on our most complicated data with randomized fault and generator parameters, denoted RANDOMIZED in Table I. The model performs surprisingly well. It learns different generator and noise parameters.

The prediction quality degrades with time and reactive power (Q) estimation is generally worse (see Fig. 8). There are ways to handle the first issue, but the second one requires an additional investigation. As already stated above, we find

TABLE I
NRMSE STATISTICS (IN %)

Model	Data	Mean	Median	95% percentile
VAR	randomized	4.72	4.59	6.52
WD-LSTM	regular	0.23	0.08	0.91
WD-LSTM	randomized	7.48	7.30	9.22
FT-WD-LSTM	randomized	0.69	0.60	1.37
FT-WD-LSTM	high-order noise	0.66	0.58	1.20

this model noteworthy due to its ability to sense, catch and benchmark non-linearities in the data. The results of the analysis are also shown in Fig. 5 in the form of a color map.

Finally, we investigate relation between magnitude of the noise magnitude and the order of the model, p. Direct application of Akaike criterion yields a counter-intuitive result that the optimal model order p increases when noise magnitude becomes smaller. We relate it to overfiting, and suggest a different method to identify the optimal order. It seems more

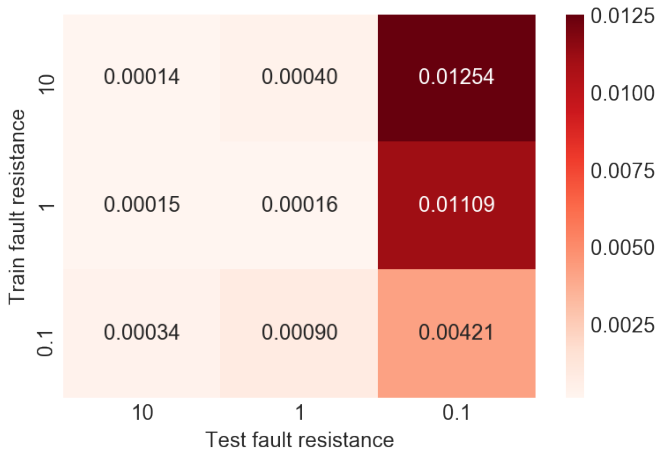


Fig. 5. Color map of the VAR performance.

appropriate to base the cruterium on the following how the learned parameter, A_i, saturates with i: $p = \arg\min_i\{\|A_i - A_{i-1}\| < \epsilon\}$. For majority of cases the optimal order of the scheme derived this way returns a sensible result, see Fig. 6.

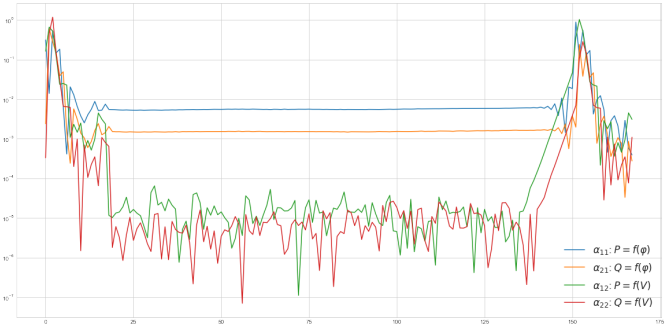


Fig. 6. Decay of the learned parameter for VAR - principal scheme to reconstruct the optimal order of the scheme from the data.

2) WD-LSTM: Let us first process NORMAL-OPERATION data correspondent to fixed generator parameters. Reconstructed and actual results shown in Fig. 9 are almost indistinguishable. The error distribution among samples in Fig. 7 is also skewed to zero.



Fig. 7. WD-LSTM: Estimated density of the NRMSE distribution.

Given that the model shows such a a good performance in the simple normal case, it is important to furhter experiment putting the model in a significantly more adversarial (towards learning) regime characterized by seious fluctuations. We train the model on the data correspondent to an additional (artificial) randomization in some parameters of the generator. This modification, as shown in Table I, allows our model to achieve quality comparable to the one provided by the original WD-LSTM model. We conclude that this architecture allows generalizations over the range of different (but reasonably close) generator parameters.

Further tests shown in Table I include learning in the regime of the HIGH-ORDER NOISE data with extremely high perturbations, but fixed (not evolving) generator parameters. Resistance of the faults is set to be three orders less than for training (this is the maximum order the generator stabilizer can handle).

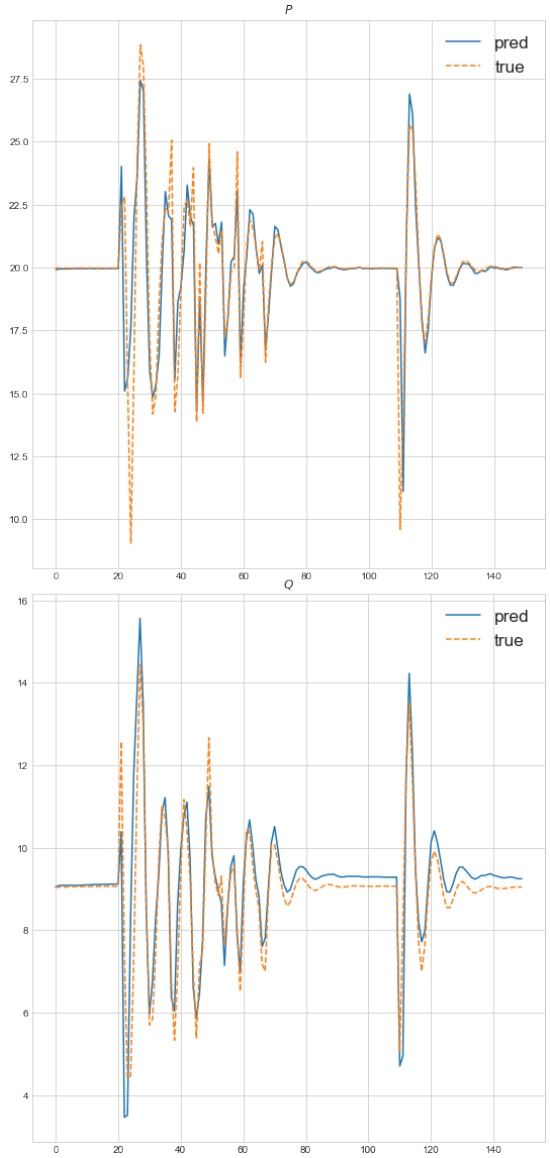


Fig. 8. VAR: predicted values vs true values.

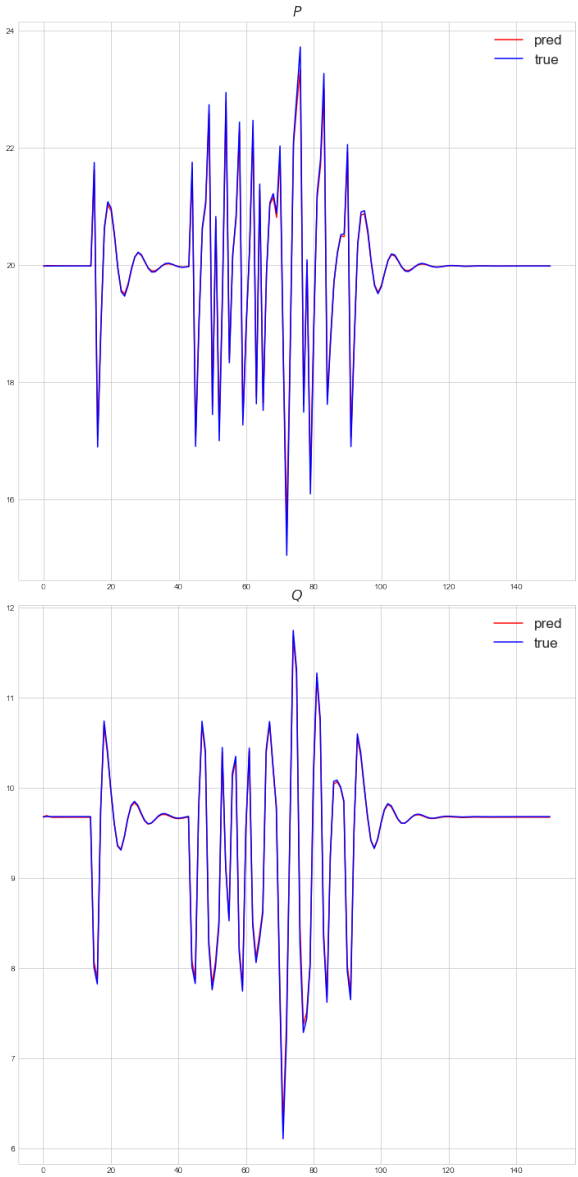


Fig. 9. WD-LSTM: predicted values vs True values.

V. Conclusions and Path Forward

In this work we have introduced a novel method, which is light and accurate for learning the model of a power generator. We have also design our custom synthetic data generation process to provide the data needed to train the models. We show that the LSTM method is truly non-linear and robust to high-order perturbations and randomization of the generator parameters, while the other, linear regression, is light in implementation but limited to regimes with small nonlinearities. We believe that such models are worth incorporating in the current grid engineering practice.

In terms of the path forward, we plan

- extending these two already calibrated approaches by a principally different one utilizing physical generator model(s);
- testing the devised algorithms on real (and not only synthetic) data;
- extend the approach to modeling power distributions, as seen aggregated on the transmission level;
- extend the approach to modeling (learning silent malfunctions as they occur) in various other precious assets of the power systems, e.g. transformers.
- integrated the developed schemes into a tool box monitoring larger power systems, including multiple generators, loads and transformers.

References

[1] P. Kundur, N. Balu, and M. Lauby, *Power system stability and control*, ser. EPRI power system engineering series. McGraw-Hill, 1994.

[2] R. P. Schulz, C. J. Goering, R. G. Farmer, S. M. Bennett, D. A. Selin, and D. K. Sharma, "Benefit assessment of finite-element based generator saturation model," *IEEE Transactions on Power Systems*, vol. 2, no. 4, pp. 1027–1033, Nov 1987.

[3] C. Lee and O. T. Tan, "A weighted-least-squares parameter estimator for synchronous machines," *Power Apparatus and Systems, IEEE Transactions on*, vol. 96, pp. 97 – 101, 02 1977.

[4] P. L. Dandeno, P. Kundur, A. T. Poray, and H. Z. El-din, "Adaptation and validation of turbogenerator model parameters through on-line frequency response measurements," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-100, no. 4, pp. 1656–1664, April 1981.

[5] M. Namba, T. Nishiwaki, S. Yokokawa, and K. Ohtsuka, "Idenntification of parameters for power system stability analysis using kalman filter," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-100, no. 7, pp. 3304–3311, July 1981.

[6] H. Lütkepohl, *New introduction to multiple time series analysis*. Springer Science & Business Media, 2005.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[8] J. Schmidhuber, "The neural bucket brigade," in *Connectionism in Perspective*, R. Pfeifer, Z. Schreter, Z. Fogelman, and L. Steels, Eds. Amsterdam: Elsevier, North-Holland, 1989, pp. 439–446.

[9] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[10] L. Vanfretti, T. Rabuzin, M. Baudette, and M. Murad, "itesla power systems library (iPSL): A modelica library for phasor time-domain simulations," *SoftwareX*, vol. 5, pp. 84 – 88, 2016.

[11] F. Milano, "An open source power system analysis toolbox," *IEEE Transactions on Power Systems*, vol. 20, no. 3, pp. 1199–1206, Aug 2005.

[12] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," *Foundations and Trends® in Machine Learning*, vol. 1, no. 1–2, pp. 1–305, 2008.

[13] H. Akaike, "Information theory and an extension of the maximum likelihood principle," in *Selected papers of Hirotugu Akaike*. Springer, 1998.

[14] G. Claeskens and N. L. Hjort, "Model selection and model averaging," *Cambridge Books*, 2008.

[15] S. Merity, N. S. Keskar, and R. Socher, "Regularizing and optimizing LSTM language models," *CoRR*, vol. abs/1708.02182, 2017. [Online]. Available: http://arxiv.org/abs/1708.02182

[16] J. Howard and S. Ruder, "Fine-tuned language models for text classification," *CoRR*, vol. abs/1801.06146, 2018. [Online]. Available: http://arxiv.org/abs/1801.06146