14th World Congress on Computational Mechanics (WCCM)
ECCOMAS Congress 2020
Virtual Congress: 11–15 January 2021
F. Chinesta, R. Abgrall, O. Allix and M. Kaliske (Eds)

SURROGATE MODELING BASED ON DYNAMIC NUMERICAL SIMULATION AND MEASUREMENTS FOR FAST EMULATION

Manuel A. Röhrl[1,2], Franz G. Listl[1,3], Veronika Brandstetter[1], Tobias Schulze[4] and Thomas A. Runkler [1,2]

[1] Siemens AG, Otto-Hahn-Ring 6, 81739 München, *manuel.roehrl@siemens.com*, *www.siemens.com*

[2] Technical University of Munich, Boltzmannstraße 3, 85748 Garching b. München, *www.in.tum.de*

[3] University of Stuttgart, Keplerstraße 7, 70174 Stuttgart, *www.uni-stuttgart.de*

[4] Siemens Energy Global GmbH & Co. KG, Freyeslebenstr. 1, 91058 Erlangen, *www.siemens-energy.com*

Key words: Machine Learning, Neural Networks, Surrogate Modeling, Hybrid Modeling, Combined Cycle Power Plant, Simulation

Abstract. Today, in many complex real-world systems, physics-based simulation models often provide sufficient precision but are computationally intensive. Machine learning surrogates, once trained, can achieve simulations by orders of magnitude faster than their original physical model without sacrificing much accuracy.

In this paper, we present a surrogate model in form of a neural network that is fitted to a set of different time series. The time series data are generated partly by a physical model and partly by measurement. This is because a physical model is available only for a part of the entire state space that is to be modeled. This method is used to predict the flue gas temperature at the output of the evaporator in the heat recovery steam generator of a combined cycle power plant. For simulation we use a specialized in house tool for transient power plant processes, called "Dynaplant". The generated surrogate model is fast and captures the major dynamics. Consequently, the model can be used in applications where fast evaluation is required, e.g., in parallel to operation. One form of such usage is virtual sensors, whereby, physical detectors can be omitted, and thus costs are reduced.

With this, we demonstrate a method that beneficially merges physical insight from simulation with real-life data and machine learning. Our findings are of interest to applications where either simulated or measured time series data or both of different operating points are available and a fast simulation model is required.

1 INTRODUCTION

Generally, fast and accurate dynamic models are required for the control and optimization of technical systems and visions, such as the digital twin [1]. One example is the modeling of transient processes in combined cycle power plants (CCPPs), see Figure 1. The ever-increasing share of renewable energies in the energy mix leads to natural fluctuations in electrical grids. These variations need to be balanced out,

e.g., by on demand operation of conventional power plants. Therefore, the demands for flexible operation are increasing steadily, and accurate simulation in plant engineering is gaining importance.

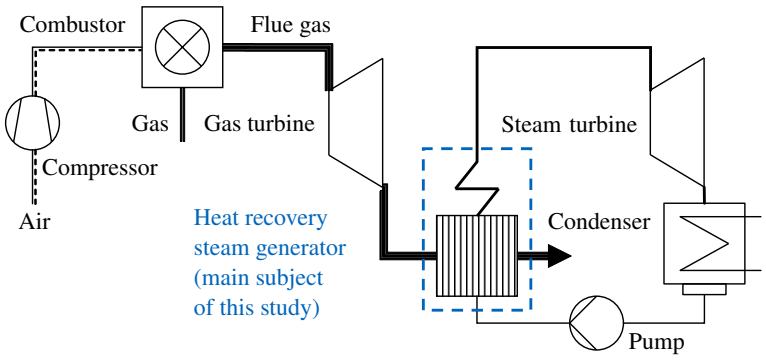


Figure 1: Simplified scheme of the CCPP Process.

However, detailed numerical simulations are computationally intensive and often unsuitable for application during operation. Hence, novel highly efficient machine learning (ML) methods are used exploit the huge amount of data acquired by such simulations.

In this study, we investigate the extent to which the data obtained from a precise numerical simulation model of a heat recovery steam generator (HRSG) and sensors can be used to train deep neural networks (NNs) or other ML models to forecast the system state, which is somewhat correlated to the model inputs. The working hypothesis is that the model learns a physical understanding of the system to generate a plausible solution for states similar to the ones from training without any further simulation effort. This data-driven solution could then be generated, orders of magnitude faster, and thus, achieve a considerable reduction in computing time compared to the numerical solution.

After this introduction, we give an overview of related work. Then, we explain the proposed methodology and discuss our results. Lastly, we summarize the results and give an outlook on future work.

2 RELATED WORK

Traditionally, dynamic processes in HRSGs have been assessed by detailed numerical simulations. By contrast, this work exploits such a model and combines it with sensor data to generate a fast emulation.

2.1 Simulations in Power Plant Engineering

Today, there are considerable number of different simulation tools for power plant design, optimization and performance prediction on component and system level. Depending on the intended purpose, one- or three-dimensional approaches and steady-state or transient simulation methods may be applied. For fluid mechanical and thermodynamic processes, the mathematical model description is based on the problem-specific formulation of balance equations for the mass, momentum, and energy. Then, the mathematical problem is solved using various numerical methods. A comprehensive overview of state-of-the-art numerical simulation methods in power plant engineering is given in [2].

In practice, dynamic simulations significantly contribute to a deeper understanding of the transient operational characteristics of a power plant. For instance, they help to develop and optimize control loops for power plant operation [3]. This way, the simulation model can facilitate smooth commissioning of the real power plant and reducing the risk of unplanned and expensive time delays.

2.2 Learning Models from Time Series Data

Research on time series forecasting has a long history. Linear statistical methods such as linear regression, Box-Jenkins and autoregressive integrated moving average have shaped time series forecasting. As a baseline, we apply linear regression to our problem. At the same time autoregressive methods were proposed, but they do not apply to our problem since our target value is not an input variable. However, in the last two decades ML methods outperformed classical approaches [4, 5]. These models can represent nonlinear relations. Our approach follows this line and applies two ML methods, which have gained prominence in recent years, namely long short-term memory (LSTM) and dual-stage attention-based recurrent neural network (DA-RNN) [6, 7, 8].

2.3 Integrating Simulation Results into ML

Numerous studies have described the augmentation of training data by making use of simulation results. Surveys, such as that conducted by von Rueden et al. give a broad overview of integrating knowledge into learning systems in general. Our approach, also called serial semi parametric hybrid modeling [9], is often employed to complement the training data with simulating scenarios not yet covered. Further, since numerical simulations are generally very complex software products requiring extensive computing time, data-driven surrogate models are often used to reduce the computation efforts [10, 11, 12].

3 METHODS

The following is a brief description of a numerical simulation model, the training data generation, and the fitting process for the surrogate model.

3.1 Modeling Dynamic Processes in the HRSG

We conducted dynamic simulations of HRSG and related systems using the in house software, Dynaplant. This is a one-dimensional numerical simulation software, which allows modeling of the physical process itself and the associated control structures. The distinguishing characteristic of the HRSG model is the detailed representation of the water/steam side in the heating surface tubes. The tube model is based on the balance equations for mass, momentum, and energy. Further, it includes proprietary know-how on heat transfer and pressure drop in two-phase flows.

Figure 2 shows a simplified sketch of the physical domain of the simulation model. Here, the investigated CCPP features the vertical, triple-pressure Benson HRSG. The hot flue gas from the gas turbine flows vertically upwards through the HRSG. On the water/steam side the incoming liquid feed water is heated up to saturation temperature, evaporated and further superheated on three pressure levels low, intermediate and high pressure to optimize the heat transfer from the flue gas to the water/steam side. While both the low- and intermediate-pressure stage feature drum-type evaporators, the high-pressure stage is equipped with a once-through Benson evaporator [3]. As an example, the heating surface design of the Benson evaporator with its meandering tubes is displayed magnified on the left side of Figure 2. The flue gas

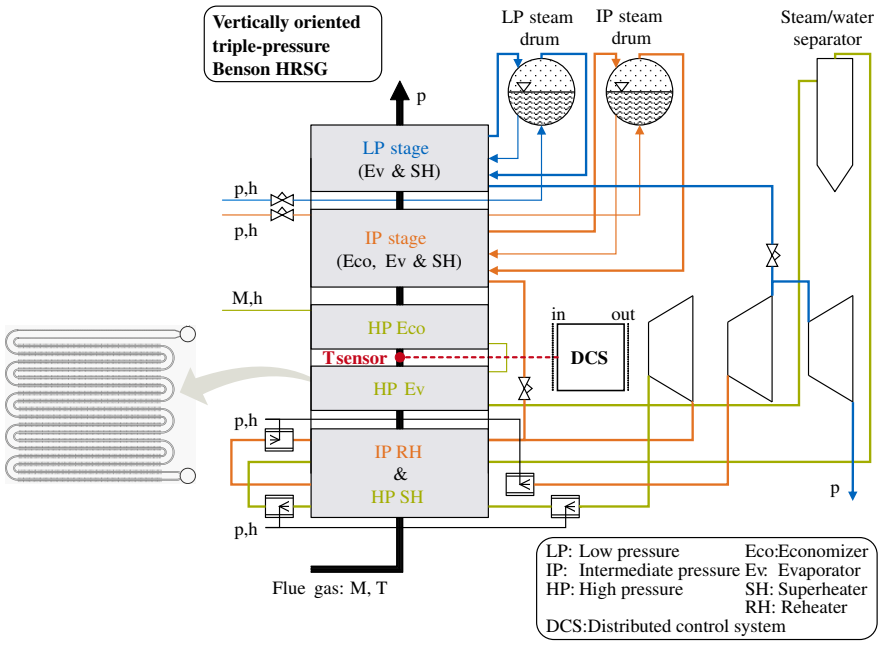


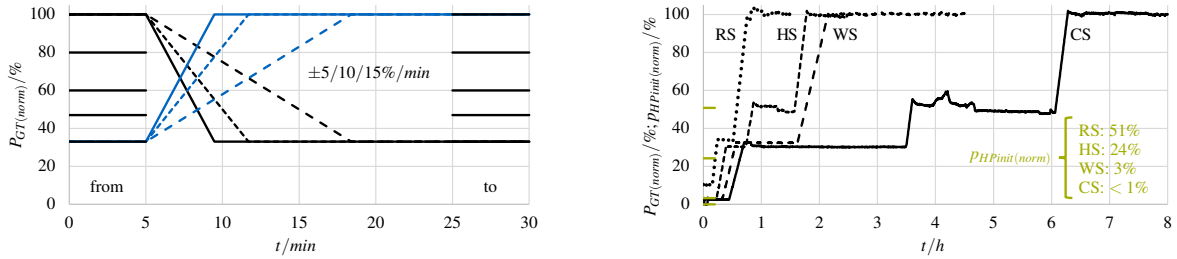
Figure 2: Schematic sketch of the numerical simulation for the transient operational characteristics of the HRSG and related systems.

temperature sensor "*T sensor*", which we want to replace with our emulation, is located in between the high-pressure evaporator and economizer-heating surfaces. Its sensor values are processed in a distributed control system to determine the instantaneous heat flow from the flue gas to the water/steam side in the high-pressure evaporator, which in turn is required to control the feed water mass flow to the evaporator.

3.2 Training Data Generation

The development of the surrogate model, i.e., the flue gas temperature emulation, is based on time series analysis of the considered flue gas temperature and correlated quantities. The data stem from two sources: First, from Dynaplant simulations of different gas turbine load ramps in normal plant operation (see Figure 3(a)). Second, from measurements in the real plant during plant start ups, i.e., cold, warm, hot, and restart (see Figure 3(b)). Notably, the application of a virtual sensor inevitably requires all data to be acquired by simulation. Some scenarios, i.e., start up procedures, just cannot be simulated with the Dynaplant model existing at the time of this study.

To generate different gas turbine load ramps, we modify three variables: the initial load, slope, and final load. The initial selection of the variables contains five values for the initial and final load and six slopes of the load change (see Figure 3(a)). This results in a total of sixty different gas turbine load ramp scenarios, which were simulated with Dynaplant. In our model, the gas turbine load ramps are realized by setting the corresponding time-dependent boundary conditions of the flue gas mass flow and temperature at the HRSG flue gas inlet (Figure 2). Due to the limited availability of measured data from plant start ups only four, representative datasets (see Figure 3(b)) could be included in the training data.



(a) Load ramps with different initial loads, final loads, and slopes.

(b) Four start ups: restart (RS), hot start (HS), warm start (WS), and cold start (CS).

Figure 3: The training data: (a) simulated load ramps and (b) measured start up procedures

However, the initial selection of training data causes two problems. First, the variables skew to certain values, so that the population of load variables is represented only to a certain extent, in addition, the initial choice of variables leads to an unnecessarily high computational effort since the simulation of one load ramp already takes about one and a half hours on an average processor unit. Second, the unbalanced ratio of the number of start up scenarios to the number of load ramps results in the former being underrepresented in the training of the NN.

To improve the selection of variables and thus reduce the simulation effort simultaneously, we use Latin hypercube sampling [13]. This ensures that data are selected from each area of the population, and at the same time randomly selected, which in general better reflects the distribution of the data. After some tests, we decided to use twenty Latin hypercube samples instead of the sixty initial samples. To address the second problem, the imbalance between the load changes and the start ups, we use a variant of stratified sampling. Here, a random number of variables were sampled with replacement from each of the two groups the simulated and the measured data so that the size of the groups was balanced. Some data points of the measured data thus entered the training loop more than once.

3.3 Building the Surrogate Model

In this study, we use a serial hybrid model structure [9]. Figure 4 shows the information flow. We use sensor data and simulation results for the surrogate model. The two data sources are used jointly as training data.

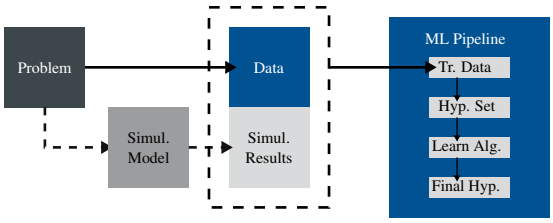


Figure 4: Information flow for the training of a surrogate model based on simulation results and measurements.

The problem of forecasting the temperature based on correlated quantities can be tackled as a problem of supervised learning. This means we learn, based on a finite set of observations, the relation between a set

of input variables and one output variable, that is somewhat dependent on the inputs. Such a mapping can then be used for one-step forecasting. In the one-step forecasting, a problem is cast in the form of a generic regression problem by creating an input data matrix,

$$X = \begin{bmatrix} x_1 & x_2 & \dots & x_n \\ x_2 & x_3 & \dots & x_{n+1} \\ \vdots & \vdots & \vdots & \vdots \\ x_{N-n-1} & x_{N-n} & \dots & x_{N-1} \end{bmatrix} \tag{1}$$

where n previous values are available and N is the number of samples. Hence, the output vector is

$$Y = \begin{bmatrix} y_{n+1} \\ y_{n+2} \\ \vdots \\ y_N \end{bmatrix}. \tag{2}$$

After this formalization of the one-step forecasting problem as supervised learning task, any regression model can be fitted to the data.

4 RESULTS AND DISCUSSION

Table 1 compares the performance of three regression models. It shows the best results after tuning the hyperparameters: batch size, learning rate, hidden size and lookback. Further, we tested different optimizers and loss functions. In addition, we applied stratified sampling to increase model accuracy. We trained the models with five-fold cross validation on 90% of the dataset described in Section 3.2, while 10% is held out for testing. This procedure was chosen because the available dataset is not particularly large.

Table 1: Comparison of different regression models.

Model	Test on load ramps		Test on start ups	
	MSE (normalized)	Variance	MSE (normalized)	Variance
Linear Regression	1,3	NA	8,22	NA
LSTM	0,0627	0,560	0,0491	9,315
DA-RNN	0,0261	0,206	0,0569	9,444

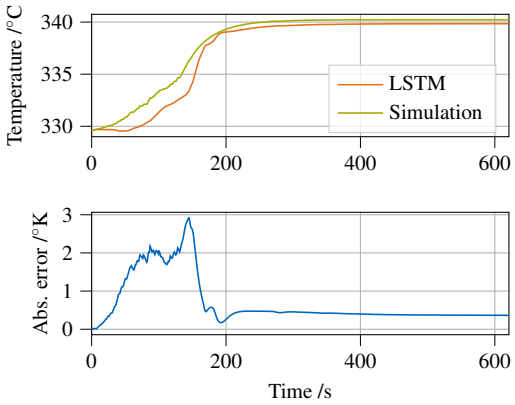
The table shows three models: a simple linear regression model for comparison, an LSTM model, and a DA-RNN model. The former cannot capture the underlying dynamics; thus, the error is large. The latter two models similarly show better performance, while the DA-RNN model can fit the load ramps better, it performs worse than the LSTM model on the start ups. Overall, the differences are marginal and in the range of the non determinism of the NNs. Since, the LSTM model is less complex and faster, it is the preferred choice for our application.

Table 2 compares the initial setting of the variables with those selected by Latin hypercube sampling within the LSTM model from Table 1. For both variants, we performed stratified sampling after simulating

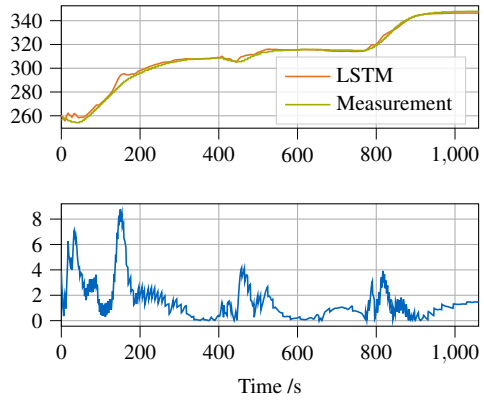
Table 2: Sampling of load ramp variables within the LSTM models.

Sampling strategy	Test on load ramps		Train. time
	MSE (normalized)	Variance	(Avg. p. Fold)
Initial setting	0,0627	0,560	413,8 s
Latin hybercube sampling	0,0205	0,0504	257,6 s

the load changes, as described above. Latin hypercube sampling is used to reproduce the parameter distribution of the load ramp variables with as few samples as possible. Testing the model trained on these samples shows that the accuracy is maintained, and the time for training is almost cut in half. This is because only twenty instead of the initial sixty simulation runs were performed when using the Latin hypercube sampling. Thus, the NN uses significantly fewer data points for training, reducing training time in addition to the originally planned reduction in simulation effort.



(a) Top: Predicted target quantity of one load ramp compared to simulation results over time; Bottom: Absolute error between both over time



(b) Top: Predicted target quantity of one start up compared to measurements over time; Bottom: Absolute error between both over time

Figure 5: LSTM model compared to one simulated load ramp (a) and one measured start up (b)

Figure 5 shows time series plots of the LSTM model from Table 1 compared to simulation results and measurements, respectively, along with as the absolute error between the two. Figure 5(a) shows one exemplary load ramp, while Figure 5(b) shows one start up procedure, the so called “warm start”. Both figures show that the overall dynamics are represented. The stationary regions are well predicted, whereas the transient sections are represented less accurately, and for short times, large errors occur.

The shown LSTM model is significantly faster in simulating one load ramp than the Dynaplant model, while the former only takes seconds the latter roughly takes one and a half hours for the same load ramp, on the same processing unit.

5 CONCLUSION

The research question in the current project sought to determine whether a surrogate model can rapidly and accurately emulate the dynamics in the HRSG of a CCPP. The results of this study indicate that this is possible. The major dynamics are represented, but the small sample size inhibited accurate modeling of transient sections.

We demonstrate an interesting concept to replace cost-intensive physical sensors using physical insight from simulation and real data to build a fast emulation. It should be mentioned, that the training data may also be acquired purely by accurate physics-based simulation. Unlike classical reduced-order models, owing to the NN approach, the physical and space-time resolution do not reduce. The study contributes to the rapidly expanding field of combining physical simulations with ML by providing an interesting example where simulation and measurement data are jointly used to train a data-driven model.

A limitation of this study is that the offline model generation required beforehand is computationally intensive since many simulation runs have to be calculated.

A bigger sample size should be adopted to answer the question of what critical amount of data is required to eliminate the current deviations. In addition, further research could usefully explore how to generalize the automatic synthetic training data generation for arbitrary problems. More broadly, further research could be conducted to determine the effectiveness of recent hybrid model architectures, such as physic informed NNs [14, 15].

ACKNOWLEDGMENT

The work this report is based on was supported with funds from the German Federal Ministry of Education and Research within the project "ALICE-III: Autonomous Learning in Complex Environments" under the identification number 01 IS 18049 A.

REFERENCES

[1] D. Hartmann and H. van der Auweraer, "Digital twins," *arXiv:2001.09747 [cs]*, Jan 2020.

[2] B. Epple, R. Leithner, W. Linzer, and H. Walter, "Simulation von Kraftwerken und Feuerungen," *Springer-Verlag*, 2012.

[3] G. Schlund, T. Schulze, F. Thomas, and J. Brückner, "New benson evaporator for vertical hrsgs exceeds expectations," *Modern Power Systems*, p. 39–41, Oct 2019.

[4] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "Statistical and machine learning forecasting methods: Concerns and ways forward," *PLOS ONE*, vol. 13, p. e0194889, Mar 2018.

[5] G. Bontempi, S. Ben Taieb, and Y.-A. Le Borgne, *Machine Learning Strategies for Time Series Forecasting*, vol. 138 of *Lecture Notes in Business Information Processing*, p. 62–77. Springer Berlin Heidelberg, 2013.

[6] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning Book*. MIT Press, 2016.

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

[8] Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang, and G. Cottrell, "A dual-stage attention-based

recurrent neural network for time series prediction," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17)*, Aug 2017.

[9] M. von Stosch, R. Oliveira, J. Peres, and S. F. de Azevedo, "Hybrid semi-parametric modeling in process systems engineering: Past, present and future," *Computers and Chemical Engineering*, vol. 60, p. 86–101, 2014.

[10] R. van der Merwe, T. K. Leen, Z. Lu, S. Frolov, and A. M. Baptista, "Fast neural network surrogates for very high dimensional physics-based models in computational oceanography," *Neural Networks*, vol. 20, no. 4, pp. 462–478, 2007.

[11] N. Ruiz, S. Schulter, and M. Chandraker, "Learning to simulate," in *International Conference on Learning Representations*, 2018.

[12] L. von Rueden, S. Mayer, K. Beckh, B. Georgiev, S. Giesselbach, R. Heese, B. Kirsch, J. Pfrommer, A. Pick, R. Ramamurthy, and et al., "Informed machine learning – a taxonomy and survey of integrating knowledge into learning systems," *arXiv:1903.12394 [cs, stat]*, Feb 2020.

[13] M. D. McKay, R. J. Beckman, and W. J. Conover, "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 21, no. 2, pp. 239–245, 1979.

[14] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019.

[15] M. A. Röhrl, T. A. Runkler, V. Brandstetter, M. Tokic, and S. Obermayer, "Modeling system dynamics with physics-informed neural networks based on Lagrangian mechanics," in *IFAC- PapersOnLine*, 2020.

A Hyperparameters

Hperparameters of final LSTM model	
Hidden layers	2 with each 16 units
Activation functions	rectified linear unit
Lookback	20
Loss	mean squared error
Batch size	16
Optimizer	Adam