Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 80 (2019) 683–688

www.elsevier.com/locate/procedia

26th CIRP Life Cycle Engineering (LCE) Conference

Machine Learning based System Identification Tool for data-based Energy and Resource Modeling and Simulation

Thomas Weber[a,*], Johannes Sossenheimer[a], Steffen Schäfer[a], Moritz Ott[a], Jessica Walther[a], Eberhard Abele[a]

[a] *Institute of Production Management, Technology and Machine Tools (PTW), Otto-Berndt-Str. 2, 64287 Darmstadt, Germany*

* Corresponding author. Tel.: +49-6151-16-26054 ; Fax: +49-6151-16-20087. *E-mail address:* t.weber@ptw.tu-darmstadt.de

Abstract

Generated machine data is often not fully utilized in modern power production, although it could provide new approaches to significantly increase productivity, flexibility and resource efficiency as well as energy efficiency in production. Data-based models, which can be created with the help of machine learning algorithms, can map the system's behavior accurately and thus provide a basis for a better system understanding for further energy und resource optimization approaches. The objective of this paper is to develop a generic system identification tool that uses the above-mentioned data-based modeling approach to optimize the electrical power and resource consumption for a given load, regardless of the considered plant or machine. Therefore, the system identification tool autonomously preprocesses the data, compares different hyperparameters for neural networks to reproduce the system's behavior and finally selects the best-suited regression algorithm with the corresponding hyperparameters.

Keywords: Energy efficiency, Ressource efficiency, System Identification, Machine Learning

1. Introduction and motivation

As the overall ecological awareness especially in western countries increases and the world primary energy consumption continues to grow in average by 1.7 % per year over the last 10 years [1], the current political and social discussions often revolve around climate change and greenhouse gas emissions. The EU climate strategy aims to reduce greenhouse gas emissions step by step until 2050, combined with an increasing share of renewable energies and improvements in energy and resource efficiency [2]. On a national level, the industrial sector represents the second-largest source of greenhouse gas emissions in Germany [3] and therefore offers great opportunities to reduce its energy and resource consumption.

Especially digitalization offers the potential to increase energy and resource efficiency in industrial production and energy supply services [4]. Industrial data management systems enable a continuous energy monitoring on machine level, provide energy forecasts and are the basis for an energy and resource optimization. Thus, there is a great need for better energy transparency through energy monitoring within industry in order to increase energy and resource efficiency in industrial production [5].

In spite of great progress of recent years in research and development, a gap exists between the available energy efficiency solutions and their actual implementation in industrial companies [6]. The barriers for the decision-making process for industrial energy efficiency are manifold. One of them is the need for simple and effectively adaptable solutions to various machine and system types. [7]

The present work introduces a machine-independent approach for modelling energy and resource demands, which is

10.1016/j.procir.2018.12.021

the basis for further optimization. In modeling, basically two approaches can be distinguished [8]:

- Physical modeling indicates the use of mathematical functions to represent physical laws
- System identification is an empirical model, based on information and data collected from input and output measurements from a system [8]

2. Background - machine learning based system identification

This paper presents an automated system identification approach, based on regression algorithms, to create transparency on energy and resource consumption of a system or a machine. In addition to the regression algorithms used, *data preprocessing* and *feature engineering* are extremely important. Existing approaches from these three areas as well as complete tools, which automatically combine these three tasks, are presented in the following section.

2.1. Automated data preprocessing

Data from real applications is usually influenced by interference factors, resulting in erroneous data that affects data mining performance [9]. Despite relatively well-monitored data collection, errors, out of range values, inconsistencies, impossible data combinations or missing values occur in the collected data or the data is not suitable for the required application [10]. For the respective data mining task to be executed as good as possible, the data must be available in the appropriate quantity, format and structure [9]. The success of machine learning algorithms essentially depends on the quality of the data they work with. They are strongly based on the respective training sets, the product of *data preprocessing*. High data quality is therefore a prerequisite for high-quality results [10,11].

Data preprocessing can be subdivided into data preparation, data reduction and partitioning. Data preparation includes the steps of data cleaning, data conversion, data integration, normalization of data, imputation of missing values and identification of noise. Data reduction covers the techniques by which a reduced representation of the original data is achieved to suit the data mining task in a proper way. Within the partitioning step, the preprocessed data is divided into a training set, which is needed to train the algorithm, a validation set, which checks the learning performance of the algorithm and the testing set, which generates the final results [12].

The amount of data generated by data reduction is reduced compared to the original data, but retains its essential structure and integrity. Data reduction includes feature selection, instance selection, discretization and feature extraction and/or instance generation [13].

Numerous algorithms already exist for automated *data preprocessing*. In [10] the most influential algorithms regarding their application, popularity and the extensions proposed in the literature are summarized. The listed algorithms treat missing values, noise filtering, instance reduction, dimension reduction, data handling for unbalanced preprocessing and discretization.

Since data preparation is an important component for the success of data mining tasks, it is essential for the correct use, that *data preprocessing* is embedded in a standardized framework.

2.2. Automated Feature Engineering

Feature engineering is a highly relevant and time consuming task in the field of machine learning [14]. With only numerical inputs being used for the regression problem in the presented work, solely mathematically derived features can be applied. The combination of features can be done by different mathematical operations, like differences, logarithms, polynomials, powers, ratios etc. [15] Also, different time lags of the input variables can be used as features to map periodic behavior of a time series or a fixed delay between input and output variables. [16] In addition, a moving average filter can be applied to the machine learning problem to be used for *feature engineering* [17]. In a further step, autoregressive moving average (ARMA) models can be used to predict a time series or a system behavior via a weighted moving average over past values of the time series as well as past forecast errors [18].

To generalize the data based model as much as possible, the most suitable parameters must be selected [12]. Within the time series analysis, the autocorrelogram serves as one of the most important tools to evaluate the realization of an output depending on past realizations [19]. With this tool, the influence of features with different time lags is investigated.

Zhang et al. examined three different approaches for feature selection for energy demand forecasting in the residential sector [20]. To evaluate individual features, they use correlation analysis, a random forest algorithm and principal component analysis methods. They conclude that even non-computing-intensive methods, such as correlation analysis, can lead to good feature selection. [20]

The mentioned *feature engineering* techniques are applied in the presented system identification tool.

2.3. Regression Algorithms

Regression is the process of learning the relationship of input space X and output space Y by adjusting parameters of a mathematical function $f: X \rightarrow Y$, so that the error between the model forecast and the real data is minimized. Often used algorithms are for instance linear regression, regression trees, support vector machine and neural networks [21]. Neural networks are widely used in the field of energy forecasting in various modifications [22–25]. Mordjaoui et al. use genetic algorithms for the parameterization of the neural network [24], although for large datasets the computation duration with this approach increases rapidly.

Alternatives for hyperparameter optimization are grid search or random search. With these approaches, a defined area of the hyperparameter space is explored. While random search examines combinations of hyperparameters randomly, grid search examines all parameter constellations with a fixed step size. The resulting parameterization can therefore be suboptimal, but the computing time is greatly reduced. However, research results have shown that the result of random search optimization is in most cases as good as a grid search [25].

Existing Tools for System identification

Diverse machine learning-based identification tools already exist with the aim of gaining new information from the available data with little effort.

AutoML is an alpha version cloud service from Google, used to generate user specific computer vision models. This is accomplished through the use of deep learning, genetic algorithms and transfer learning, which allows to achieve good results even for a small amount of user data. This is feasible because Google uses a huge set of images to pretrain its network architecture. Accordingly, AutoML is not well suited in fields, where little data is available. Therefore, the focus of AutoML currently lies in computer vision and not in the analysis of industrial data. [26,27]

TeaPot is an OpenSource Python Library, which enables automated machine learning for classification or regression problems. Similar to AutoML the core idea of TeaPot is to automate the *data preprocessing*, *model selection* and hyperparametrization of the models. The user only has to specify the accuracy of the model, instead of having actual knowledge about his data. However, TeaPot is bound to a fixed spectrum in the possibilities of *data preprocessing* and *model selection*. It is neither able to perform problem-specific *feature engineering* nor to parameterize complex neural networks [28].

The presented system identification approach pursuits the goal to be as generic as possible on the one hand, and to be focused on the field of industrial energy data on the other hand. Thus, technical expert knowledge of industrial system data is embedded in the system identification tool, but is not required to create user-defined models.

3. System identification tool for databased energy and resource modeling

The system identification tool creates data-based models on the basis of regression algorithms used in machine learning to model the real behavior of a machine or a system, based on the data of the system's input and output variables.

Figure 1 shows the program flow chart of the system identification tool. In a first *data preprocessing* step, the provided raw data is cleaned and prepared. In a next *feature engineering* step, relevant features are created and selected, as described in section 3.2. Then, the prepared data is divided into a training, a validation and a testing data set. This data is used to construct various data-based models, which is explained in section 3.2. With the training data, neural networks are trained for different hyperparameters. Finally, the regression algorithm with the lowest root mean squared error (RMSE) is selected via cross-validation. The final system identification model is used for the intended application to make new predictions.

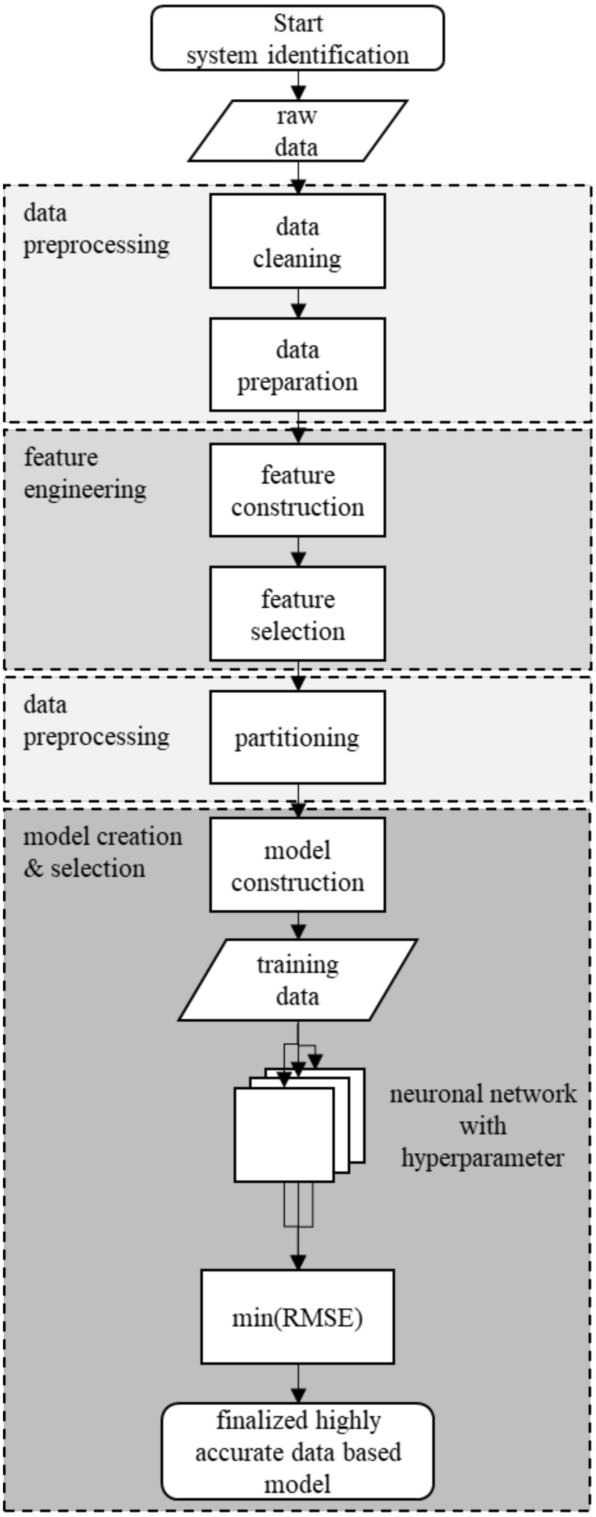

Fig. 1. Program flow chart of the system identification tool

3.1. Automated data preprocessing and feature engineering for industrial system data

In a first step, the data to be learned is cleaned up with regard to its missing values. In addition, the input values are centered by the median and the data is scaled according to its quantile range.

According to Fig. 1, *feature engineering* is divided into feature construction and feature selection. In the feature construction step, new time series are formed based on the input series of the analyzed system. Therefore, a wide range of additional data is generated. As explained in section 2.2, the following commonly used *feature engineering* methods can provide additional information to the machine learning problem:

- Time offset to the input series
- Mean value filtering of the input series
- Time-shifted mean value filtering of the input series

For the time offset to the input series, the original data is shifted in steps of 10 up to 100 time steps, which allows to take delays between input and output features into account. A moving average filter is then applied to the original data as well as the time-shifted data with window-sizes from 15 to 60 time steps and a step-size of 15 time steps. This allows the state of a system to be mapped and the dependency on the average previous mode of operation to be displayed.

In the feature selection step, the generated features are then evaluated using both, univariate linear regression and a gradient boosting regression tree.

3.2. Data based model creation and selection

As shown in section 2.3, different regression algorithms can be used to create the data-based model. The main differences of these *model creation* algorithms are the robustness, accuracy and computing time for regression. As the model needs to be trained only once, the computing time is of minor importance. The presented system identification tool focuses on neural networks, as they are a commonly used and highly robust regression method.

Neural networks can be parameterized with many different hyperparameters. Due to the large number of possible hyperparameters, the selection of a suitable network structure is implemented using a random search algorithm. In a first step, the number of neurons used is varied in steps of 10, using different epochs, batch sizes and dropout values. A detailed view of the parameters is given in Table 1. The model with the least mean squared error for predicting the test data set is selected and can be used for further applications.

Table 1. Possible hyperparameters in the system identification tool

Hyperparameter	Options
Number of hidden layers	[1; 2; 3]
Number of neurons per layer	[10; 20; 30; 40; 50; 60; 70; 80; 90, 100]
Number of epochs	[1; 2; 5; 10]
Batch size	[50; 100; 200]
Dropout rate	[0.0; 0.1; 0.3]

The number of hidden layers and the number of neurons per hidden layer determine the complexity of the model. By varying between 1 and 3 hidden layers and between 10 and 100 neurons, both simpler and more complex input and output data can be mapped. The number of epochs determines how often a dataset is passed through the learning algorithm. Especially in smaller datasets, it can be helpful to use the same data more than once for the training process. Since the entire data set cannot be learned at once, it is divided into smaller sections known as batches. Also for the number of epochs as well as the size of the batches, a wide range of options for hyperparameter optimization were chosen. The dropout rate describes the probability with which individual neurons are omitted when training the network with a batch. This can lead to much more robust nets and therefore better results.

4. Deployment of the automated system identification tool

The advantage of this data based system identification approach compared to physical modelling is the low effort and the rapidity with which a real system can be modelled. Furthermore, a not quite as profound understanding of the system is required, as it would be the case when applying a physical model. Whenever the input and the output parameters are known and their data is sufficiently well available, the tool can represent the actual system behavior in a highly accurate way. The system identification tool developed in this paper can be used for various modelling tasks like for example

- transparency on the energy and resource consumption of a system,
- condition monitoring,
- load forecasting or
- validation of optimization models.

For creating transparency on a system's energy and resource consumption, the system's inputs can be correlated to its energy and resource consumption data. There are already approaches which apply system identification to a power disaggregation problem in order to increase energy transparency [29]. System identification can be used for condition monitoring and fault detection of various different types of systems, like wind turbines [30], transistors [31] and power systems [32]. Also system identification can be applied to forecast the energy demand of buildings [33] and cities [22].

5. Application and validation of the system identification tool for a cogeneration unit

The presented system identification approach was tested on the basis of real data from a cogeneration or combined heat and power (CHP) unit. In the first step, the system identification tool was trained with the data of the CHP unit, in order to validate the quality of the achieved data-based model with a test data set. The system used is a gas-powered cogeneration unit, type "Viessmann - Vitobloc EM 5/16".

Within the CHP unit, natural gas is burned, which generates waste heat, that is transferred to a water circuit, and electrical power P_{el}. The water circuit has a certain volume flow $\dot{V}_{\mathrm{water}}$, which transports the fluid with the warmed up inlet-flow temperature T_{inlet} to heat sinks and returns to the CHP unit with a cooled return-flow temperature T_{outlet} . Thus, the cogeneration unit is characterized by the following inputs:

- electrical power generation status setting Set_P_{el}
- input temperature of the heat-conducting water T_{inlet}
- volume flow of the heat-conducting water $\dot{V}_{\mathrm{water}}$,

as well as the following predicted system outputs:

- outlet temperature of the heat-conducting water T_{outlet}
- generated electric power P_{el}

Figure 2 shows a schematic illustration of the inputs and outputs of the co-generation unit. The measured variables and their corresponding measurement accuracy are summarized in Table 2.

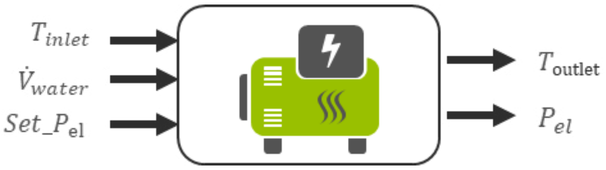

Fig. 2: Schematic illustration of the inputs and outputs of the cogeneration unit

Table 2. Overview of the measured variables and their measurement accuracy

Measured variable	Accuracy of the measurement
Electrical power generation status setting	± 0 %
Inlet temperature	± 2 %
Volume flow water	± 15 %
Electric power	± 1 %
Outlet temperature	± 2 %

The system identification tool is trained with more than 100,000 data samples, according to the procedure presented in Figure 1, consisting of data preprocessing, feature engineering, as well as model creation and selection. The hyperparameters of the neural network with the best approximation are shown in Table 3. Here, a root mean squared error (RMSE) of 56.6 W or 0.7 % mean percentage error could be achieved for the predicted values of the electric power consumption. The prediction of the thermal outlet temperature has a RMSE of 0.53 °C.

The distribution of the prediction error for the outlet temperature and the generated electrical power for the best fitting network configurations are shown in Figure 3. While the neural network provides a very good approximation of electrical power, the outlet temperature still shows a significant error due to thermodynamic laws. The outlet temperature depends strongly on the water volume flow, whose measurement error is high. This makes an accurate prediction of the outlet temperature much more difficult.

Table 3. Best matching hyperparameters of the neural network for the considered dataset

Hyperparameter	electrical power	outlet temperature
Number of hidden layers	1	1
Number of neurons per layer	50	20
Number of epochs	10	10
Batch size	50	50
Dropout rate	0.1	0.0

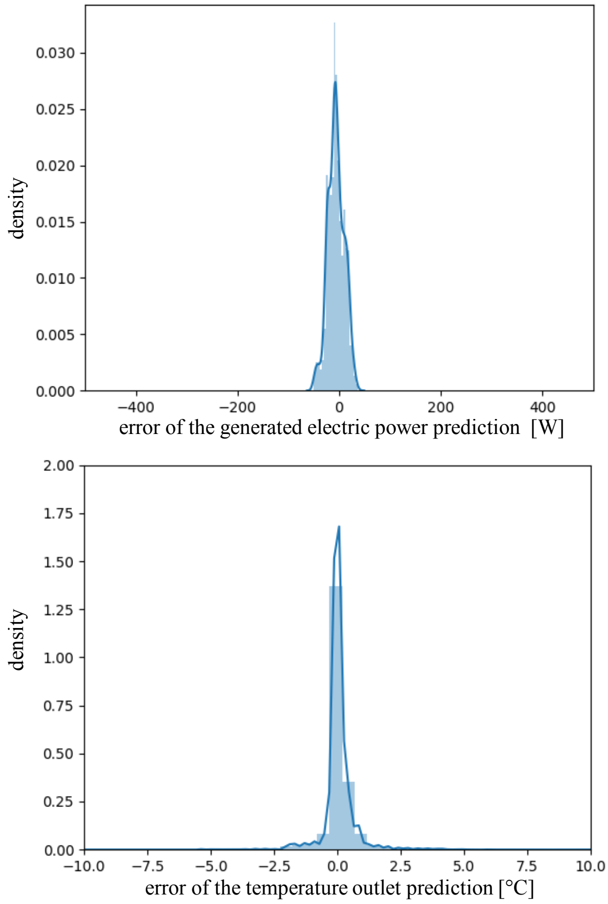

Fig. 3. Probability density function of the prediction errors of the temperature outlet (upper diagram) and the generated electric power (lower diagram)

6. Conclusion

In this work, an automated system identification tool is presented, which maps the system behavior based on measured input and output data of a system. The tool creates highly accurate data-based models for the energy and resource consumption of a machine or a system. Due to the automated *feature engineering* and network configuration, only little effort and relatively short set-up time is needed, compared to physical modelling. In addition, the tool requires only the knowledge of the system's input and output parameters as well as a sufficient quality of the corresponding data.

The tool was applied to a CHP unit, where a highly accurate representation of the system's behavior could be obtained. In a next step the tool has to be validated with further industrial use cases in order to demonstrate the generalizability of the approach. Other application scenarios besides the creation of transparency on the energy and resource consumption of a system are condition monitoring, load forecasting and the validation of energy and resource optimization models. Additionally the performance and the accuracy limits of the system identification tool need to be evaluated compared to a manually supported modelling process.

Acknowledgements

The authors gratefully acknowledge the financial support of the Kopernikus-Project "SynErgie" by the Federal Ministry of Education and Research (BMBF) and the project supervision by the project management organization Projektträger Jülich (PtJ).

References

[1] BP Statistical Review of World Energy (2018) *BP Statistical Review of World Energy 2018*. 67th ed., London.

[2] European Comission (2012) *Energy: Roadmap 2050*. Publications Office of the European Union, Luxembourg.

[3] Federal Ministry of the Environment, Nature Conservation and Nuclear Safety (BMU). *Climate Action Report 2016: On the German government's Climate Action Programme 2020*. https://www.bmu.de/fileadmin/Daten_BMU/Pools/Broschueren/klimaschutzbericht_2016_en_bf.pdf (accessed on 17.08.2018).

[4] Federal Ministry of the Environment, Nature Conservation and Nuclear Safety (BMU) (2016) *Climate Action Plan 2050: Principles and goals of the German government's climate policy*, Berlin.

[5] O'Rielly K, Jeswiet J (2015) The Need for Better Energy Monitoring within Industry. *Procedia CIRP* 29:74–9.

[6] Bunse K, Vodicka M, Schönsleben P, Brülhart M, Ernst FO (2011) Integrating energy efficiency performance in production management – gap analysis between industrial needs and scientific literature. *Journal of Cleaner Production* 19(6-7):667–79.

[7] Trianni A, Cagno E, Farné S (2015) Barriers, drivers and decision-making process for industrial energy efficiency: A broad study among manufacturing small and medium-sized enterprises. *Applied Energy* 162:1537–51.

[8] Ljung L, Glad T (2016) *Modeling & Identification of Dynamic Systems*. Studentenliteratur AB, UK.

[9] Pyle D (2005) *Data preparation for data mining*. Morgan Kaufmann, San Francisco, Calif.

[10] García S, Luengo J, Herrera F (2016) Tutorial on practical tips of the most influential data preprocessing algorithms in data mining. *Knowledge-Based Systems* 98:1–29.

[11] Kotsiantis S, Kanellopoulos D, Pintelas P (2006) Data Preprocessing for Supervised Learning. *International Journal of Computer Science* 1:111–7.

[12] Marsland S (2015) *Machine Learning: An algorithmic perspective*. CRC Press, Boca Raton, FL.

[13] García S, Luengo J, Herrera F (2015) *Data Preprocessing in Data Mining*. Springer International Publishing; Imprint; Springer, Cham.

[14] Jason Brownlee. *Discover Feature Engineering, How to Engineer Features and How to Get Good at It*. https://machinelearningmastery.com/discover-feature-engineering-how-to-engineer-features-and-how-to-get-good-at-it/ (accessed on 01.10.2019).

[15] Heaton J (2017) *An Empirical Analysis of Feature Engineering for Predictive Modeling*.

[16] Li C, Ding Z, Zhao D, Yi J, Zhang G (2017) Building Energy Consumption Prediction: An Extreme Deep Learning Approach. *Energies* 10(10):1525.

[17] Chihli Hung, Chih-Neng Hung, and Szu-Yin Lin (2014) Predicting Time Series Using Integration of Moving Average and Support Vector Regression. *International Journal of Machine Learning and Computing*(6):491–5.

[18] Lemke C, Gabrys B (2010) Meta-learning for time series forecasting and forecast combination. *Neurocomputing* 73(10-12):2006–16.

[19] Chatfield C (2004) *The analysis of time series: An introduction*. 6th ed. Chapman & Hall/CRC, Boca Raton, Fla.

[20] Zhang C, Cao L, Romagnoli A (2018) On the feature engineering of building energy data mining. *Sustainable Cities and Society* 39:508–18.

[21] Stulp F, Sigaud O (2015) Many regression algorithms, one unified model: A review. *Neural networks the official journal of the International Neural Network Society* 69:60–79.

[22] He W (2017) Load Forecasting via Deep Neural Networks. *Procedia Computer Science* 122:308–14.

[23] Chen C, Liu Y, Kumar M, Qin J (2018) Energy Consumption Modelling Using Deep Learning Technique — A Case Study of EAF. *Procedia CIRP* 72:1063–8.

[24] Mordjaoui M, Haddad S, Medoued A, Laouafi A (2017) Electric load forecasting by using dynamic neural network. *International Journal of Hydrogen Energy* 42(28):17655–63.

[25] Bergstra J, Bengio Y (2012) Random Search for Hyper-Parameter Optimization. *Journal of machine learning research*(13):281–305.

[26] Real E. *Using Evolutionary AutoML to Discover Neural Network Architectures*. https://ai.googleblog.com/2018/03/using-evolutionary-automl-to-discover.html (accessed on 18.09.2018).

[27] Zoph B, Vasudevan V, Shlens J, Le Q. *AutoML for large scale image classification and object detection*. https://ai.googleblog.com/2017/11/automl-for-large-scale-image.html (accessed on 18.09.2018).

[28] Olson RS, Bartley N, Urbanowicz RJ, Moore JH (2016) Evaluation of a Tree-based Pipeline Optimization Tool for Automating Data Science. In: Neumann F, Friedrich T, Sutton AM, (Eds.). *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference - GECCO '16*. ACM Press. New York, New York, USA, pp. 485–492.

[29] Panten N, Abele E, Schweig S (2016) A Power Disaggregation Approach for Fine-grained Machine Energy Monitoring by System Identification. *Procedia CIRP* 48:325–30.

[30] Cross P, Ma X (2014) Nonlinear system identification for model-based condition monitoring of wind turbines. *Renewable Energy* 71:166–75.

[31] Wang C, Ji B, Song X, Pickert V, Cao W IGBT condition monitoring with system identification methods. *2014 IEEE Conference and Expo (2014)*, pp. 1–6.

[32] Pierre JW, Trudnowski D, Donnelly M, Zhou N, Tuffner FK, Dosiek L (2012) Overview of System Identification for Power Systems from Measured Responses. *IFAC Proceedings Volumes* 45(16):989–1000.

[33] Li X, Wen J, Bai E-W (2015) Building energy forecasting using system identification based on system characteristics test. *2015 Workshop on Modeling and Simulation of Cyber-Physical Energy Systems (MSCPES)*. IEEE, pp. 1–6.