Combining Machine Learning and Optimization for Efficient Price Forecasting

Arild Helseth
SINTEF Energy Research
Trondheim, Norway
arild.helseth@sintef.no

Eivind Bekken Sveen
Norwegian University of Life Sciences (NMBU)
Ås, Norway
eivind.bekken.sveen@nmbu.no

***Abstract*—We present a framework based on machine learning for reducing the problem size of a short-term hydrothermal scheduling optimization model applied for price forecasting. The general idea is to reduce the optimization problem dimensions by finding patterns in input data, and without compromising the solution quality. The framework was tested on a data description of the Northern European power system, demonstrating significant reductions in computation times.**

***Index Terms*—Power system economics, Machine learning, Linear programming.**

I. Introduction

A. Scope

Electricity markets in most regions of the world will see a higher share of generation from intermittent and renewable energy sources in the future. Short-term uncertainties are increasingly likely to impact system operation, which in turn will impact the volatility of electricity prices. Fundamental electricity price forecasting models will thus benefit from a higher level of technical detail and a finer time-resolution to capture the volatility and provide accurate forecasts.

With increased model complexity and finer time-resolution, the computational times of fundamental market models will increase. This work investigates how the complexity of the short-term hydrothermal scheduling (STHTS) model presented in [1] can be reduced through machine learning (ML). By using ML to exploit patterns in the input data, one can reduce the size of the optimization model before solving it, and thus save computation time.

B. Literature Review

The use of fundamental optimization models for forecasting power prices is a well-established practice in many power markets [2]–[4]. The fundamental models serve to explain market prices from the marginal generation costs, without necessarily representing all details of the underlying market structure. Several articles have explored the use of ML in electricity price forecasting, and a recent overview is given in [5]. The models based on ML are mainly concerned with statistical modelling of consumer prices and not fundamental modelling approaches. ML has also been extensively used in energy planning models for forecasting energy demand and consumer load [6]. An extensive literature review of long- and mid-term electricity price forecasting models is provided in [7].

Several recent works have proposed techniques for reducing the electricity market problem size in the temporal dimension [8]–[10]. In [8] an approach to aggregate time steps based on minimizing the sum of absolute gradients of the residual load is presented. The authors in [9] analyze the input data of an electricity market model to dynamically identify a time-step aggregation that has limited impact on the optimization results. A method based on hierarchical clustering of consecutive hours according to conventional distance measures is proposed in [10]. Problem size reduction in the spatial dimension has been investigated for decades, i.e., by aggregating detailed hydropower plants and reservoirs into equivalent representations [11]–[13]. Automized procedures for such aggregation has been proposed by [14], [15]. To this end, we note that little work has been reported on *combining* ML and fundamental market modelling to arrive at simplified fundamental optimization models.

C. Contribution

This work is a summary of the M.Sc. thesis in [16]. We study a STHTS scheduling model based on linear programming (LP) and present a framework for reducing the LP problem sizes. For a given LP problem instance, low-impact constraints are identified and used as target data to train a ML model. The framework was tested using an artificial neural network (ANN) to predict low-impact constraints, based on input data on inflow, wind power production and demand. The predicted constraints were then removed, and the reduced LP problems were solved to obtain the price forecast.

More than 10000 LP problems were generated by a STHTS model in [1] and studied by using the proposed framework. It was found that the average computation time can be reduced by 55%, while the mean percentage error in total system cost was 0.14%.

II. Model and Input Data

A. STHTS Model

This section provides a brief mathematical description of the STHTS model, for a more detailed description, see [1], [17]. The model finds the optimal short-term operation of

This work was funded by The Research Council of Norway Project No. 268014

This is the accepted version of an article published in 2020 17th International Conference on the European Energy Market - EEM
http://dx.doi.org/10.1109/EEM49802.2020.9221968

a hydrothermal system, provided that all marginal costs and system parameters are perfectly known for the next 24 hours. An hourly time-resolution is used. The model is formulated as an LP problem in (1),

$$\min\ c^{\mathsf{T}} x \tag{1a}$$
$$Ax = b \quad (\lambda) \tag{1b}$$
$$x \geq 0 \quad , \tag{1c}$$

where c is the cost vector, x the vector of decision variables, A the coefficient matrix, and b the vector of constraint right-hand side parameters. In this work we are primarily interested in the price vector λ. The objective (1a) is to minimize the costs of operating the system, mainly comprising the thermal generation cost, the purchase of power and curtailment of price-inelastic demand. A variable approximating the expected future cost of operating the system as a function of water stored in the reservoirs is also included [18].

The major constraints per time step (1 hour) represented in (1b) are listed below:

- Hydropower reservoir balances, including detailed water-course topologies
- Power balances for each price zone, where exchange between zones is limited by grid topology and capacities
- Start-up costs on thermal power plants
- Ramping constraints on HVDC cables
- Benders cuts approximating the future cost function

All measures for efficiency of the problem reduction discussed in the remainder of this work relates to the cost of operating the system ($c^{\mathsf{T}} x$). Thus, we make the rather strong assumption that if the optimal cost does not change much when reducing the LP model, prices will not change much either.

B. Data

We applied the STHTS model on a set of data instances derived from a description of the Northern European power system, as illustrated in Fig. 1. The dataset comprises more than 1000 hydropower stations. A detailed description of an updated version of this dataset is provided in [19].

Apart from the static data description, time-varying data of load, inflow and wind power was created as follows:

- `Load` with hourly time-resolution was generated based on common power consumption pattern, and then adjusted to historical temperature data.
- `Inflow` with daily time-resolution was obtained from historical records.
- `Wind power` with daily time-resolution was obtained from historical records.

III. Methodology

The optimal solution of the LP problem (1) will be located at an extreme point of the feasible region. The feasible region takes the form of a convex polyhedron defined by the constraint set (1b)-(1c). Both the simplex and the dual simplex algorithms iterate through a subset of the edges of the feasible region, until an optimal solution is found. By removing constraints that do not constrain the optimal solution, the overall computational complexity will in most cases be reduced.

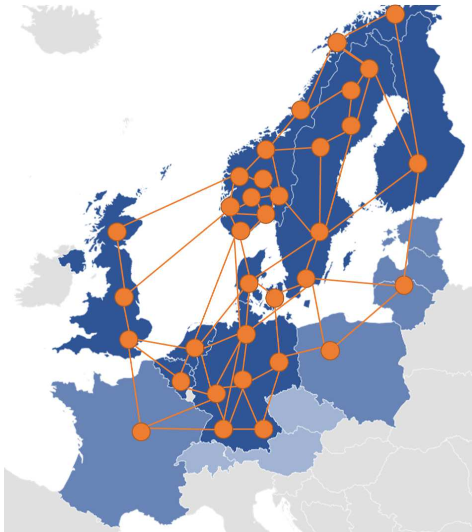

Fig. 1. Illustration of price areas and system boundaries in case study. Darker (blue) color indicates more detailes system representation.

How can we know if certain constraints do not significantly impact the optimal solution *before* solving the optimization problem? Most LP solvers have pre-processing functionality to remove redundant constraints before solving the problem. We assume this step has already been conducted. In addition, for our problem described in the previous section, some types of constraints can easily be removed a priori based on problem-specific knowledge. As an example, reservoir balances for large reservoirs can often be omitted. Typically, these reservoirs start the day with volumes well within the allowed boundaries and cannot possible reach their boundaries within the day. In this work we go a step further, using ML to remove constraints that are likely not to contribute significantly to the cost of optimally operating the system. The methodology consists of a problem size reduction method described in Section III-A, a constraint classification scheme described in Section III-B and finally the ML method based on ANN described in Section III-C.

A. Preparing Training Data with a Genetic Algorithm

A Genetic Algorithm (GA) was used to provide the ML algorithm with training data, in the form of reduced LP models. We introduced a binary variable to represent each constraint in (1b), and let the GA control if the constraint should be included or not. After fixing these binary variables, a modified set of constraints replaces the original constraints in (1b) before solving (1). The modification consisted in introducing a slack variable for each of the deactivated constraints. The GA was implemented using DEAP, which is a framework written in Python for performing evolutionary computations [20].

This is the accepted version of an article published in 2020 17th International Conference on the European Energy Market - EEM
http://dx.doi.org/10.1109/EEM49802.2020.9221968

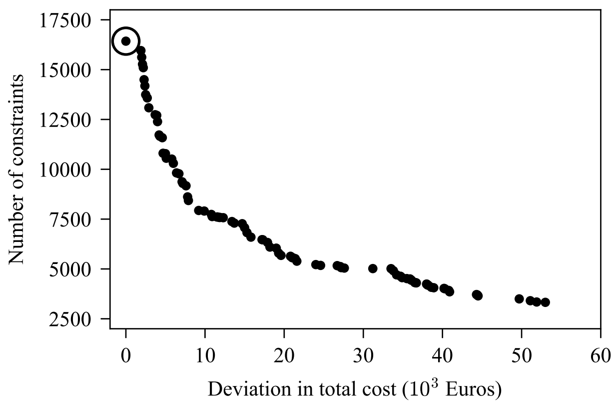

Fig. 2. Pareto front obtained from the GA model reduction.

Fig. 2 illustrates the Pareto front, mapping the minimum deviation in system cost, i.e. the objective value in (1a), for a given number of constraints. After running the GA, we are left with a large number of LP models (or set of active constraints) along the Pareto front. These have now been efficiently reduced in the two considered dimensions.

By analyzing the type of constraints being removed, we can also obtain important insight on the importance of certain constraint types. As an example, the majority of the removed constraints were water balances for larger reservoirs. As mentioned earlier, this particular type of constraint can also be removed a priory. This was done in the circled point with zero cost deviation in Fig. 2.

B. Constraint Classification

One challenge in predicting the constraints that can be removed is the large number of individual constraints that must be classified. The STHTS problem typically has several million constraints. Creating a multi-label model, with each label representing a constraint, is challenging both from a computational and statistical point of view. To overcome this problem, we split the label-space into smaller parts, creating classification models for specific types of constraints. In this work, the following six constraint labels were used:

- Water balances
- Discharge balances
- HVDC cable ramping
- Reservoir bounds
- Bypass bounds
- Discharge bounds

The major advantages of making label clusters are the gain in computational speed and the interpretability of the results. On the negative side, one runs the risk that relations between different parts of the original model may not be discovered by the ML model.

In this work, several simplifications were made to handle the vast amount of data. First, the LP models were simplified assuming imports and exports of power from exogenous markets as constant values, and that constraints on bypass and

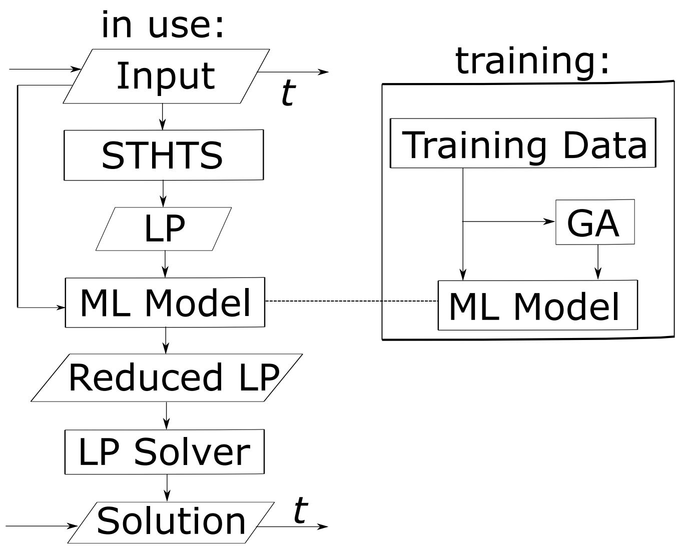

Fig. 3. Flow chart illustrating the use of ML to reduce LP size.

reservoir were not subject to seasonal changes. Thus, only the right-hand sides of the water- and power balances were allowed to vary with time, and none of the variable bounds were changed. When training the model, only variations in inflow, load and wind power were considered, as discussed in Section II-B.

C. Using ML to Reduce Problem Size

An ML model based on ANN was trained to find patterns that can inform when constraints should be deactivated. The framework for integrating optimization and ML is illustrated in Fig. 3.

The STHTS model formulation for a given time instance serves as a starting point. Subsequently the GA described in Section III-A is run to create training data. The targets used for training the ML model are binary vectors indicating whether a constraint is deactivated or not. The targets were computed in two consecutive steps:

1. Solving the LP problem, and then evaluating the dual values to find which constraints can be removed.
2. Use the proposed GA to reduce the problem size even further.

Step 1 was found necessary from a computational perspective, to reduce the size of the population to be evaluated by the GA. We note that the importance of conducting step 1 depends on the compactness of the initial model formulation.

The same input data is provided to the STHTS and the ML model, as illustrated in Fig. 3. Thus, the ML model identifies constraints that can be deactivated independently of the solution from the STHTS model. Based on the information gained from the ML model, the LP problem is reduced before being fed to the LP solver.

The ANN model was implemented with Keras [21] and TensorFlow [22]. The main goal is to find the accuracy of the reduced LPs suggested by the ML model, and the

This is the accepted version of an article published in 2020 17th International Conference on the European Energy Market - EEM
http://dx.doi.org/10.1109/EEM49802.2020.9221968

corresponding change in computational time. The accuracy was measured as the mean absolute error in system objective value, and the change in computational time as the percent deviation between solution times of the initial and the reduced LP problem.

Before training the model, the data were randomly shuffled, and 10% of the data was reserved as a test set to be used for evaluating the performance of the final model. The remaining data were further split into 4 equally sized folds to be used for cross-validation during the training of the model.

The ANN was made with 4 fully connected neural layers. The data were first fed into 3 intermediate neural layers that used a rectified linear unit activation defined as $f(z) = \max(0, z)$. The intermediate layers had up to 2 times more units than the number of input units to ensure that the model could be able to capture sufficient structure in the underlying data. The output layer used the sigmoid activation, given by $\sigma(z) = 1/(1+\exp(z))$, and yields likelihoods for which class each label belong to. In addition, dropout layers with a dropout rate of 0.5 were placed between each neural layer to combat overfitting [23]. A dropout rate of 0.5 means that half of the inputs to the next layer are chosen at random and set equal to zero, which forces the model to learn redundant patterns, creating a more robust model.

The loss function that was optimized during training is based on binary cross-entropy. To further combat overfitting, we used L2 regularization in each neural layer. The loss-function is optimized with the RMSProp optimizer [24], which improves upon earlier stochastic gradient descent methods.

IV. Case

A. ML performance

Fig. 4 shows graphs of the receiver operating characteristics (ROC) for the 6 constraint labels when applied to the dataset. The figure shows both the micro and macro-averaged curves, that are obtained by averaging the true and false positive rates of each label. The diagonal straight dotted line indicates the expected performance over time by randomly guessing each label. The micro-averaged ROC curves for all 6 models in Fig. 4 tells that the model is able to predict the most common labels quite well, with micro-averaged area under the curves (AUC) scores ranging from 0.86 to 0.94.

In Fig. 5, the total system cost for the original and the predicted model are presented. At low system costs, the modified model underestimates the system costs, indicating that the false positive ratio is too high and that too many constraints have been removed. When the system costs are higher, the system costs of the reduced models deviate little from the system costs of the original problems. We did not further investigate the reasons for this systematic difference.

Fig. 6 shows the changes in computation time and the number of Simplex iterations when reducing the LP problem size. The computation time is measured in deterministic ticks, which is a unit provided by CPLEX to measure the workload. The average computation time was reduced by 55%. Although

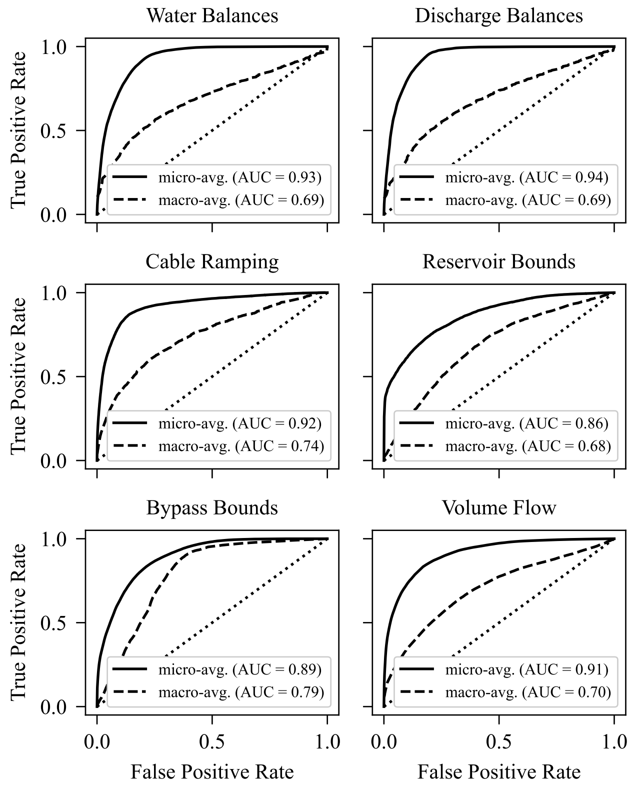

Fig. 4. ROC curves related to the test performance of the ML models. The solid curves are the micro averages, and the dashed curves are the macro averages.

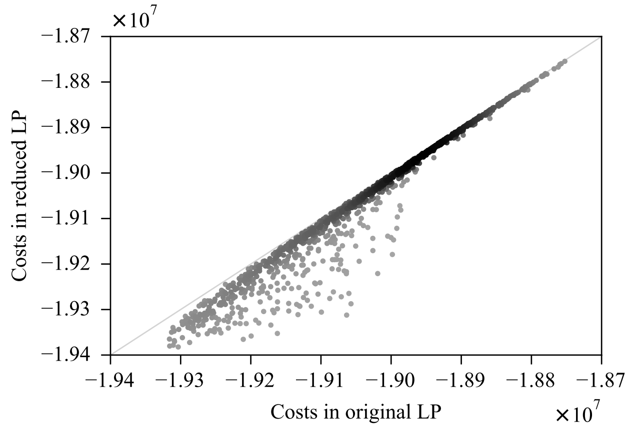

Fig. 5. Scatter plot showing the costs of the original problem, and of the corresponding reduced problem (after deactivating reservoir balances and cable constraints as suggested by the ML model). The test set consist of randomly selected daily problems.

This is the accepted version of an article published in 2020 17th International Conference on the European Energy Market - EEM
http://dx.doi.org/10.1109/EEM49802.2020.9221968

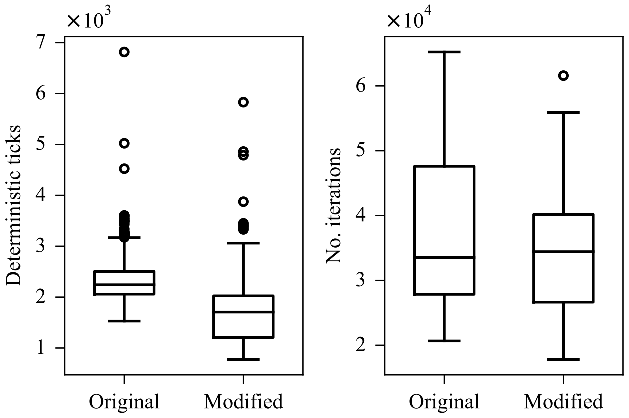

Fig. 6. The distribution of computation times (left) and the number of iterations (right) for the original and the reduced problems. The black boxes inside the plots describe the interquartile range, horizontal bars indicate the mean values.

the computation times are reduced, the average number of Simplex iterations increased by 12%.

V. Conclusions and Future Work

We presented a ML-based framework for problem size reduction of a short-term hydrothermal scheduling optimization model. On a general note, designing efficient mathematical models is an art, and most models have room for improvement. Some of these improvements can be harvested through careful model design, including only variables and constraints that are strictly necessary. Moreover, domain knowledge can be used to include constraints only when needed (relaxation). Still, after such improvements, this work demonstrates ample capability of problem reduction through ML without significantly reducing the solution quality.

Finally, we would like to point out some possible paths worthwhile further exploration:

- Rigorous testing and verification of the type and number of classes used in the multi-label classification problem.
- We applied the objective function value as the performance measure for the ML, while the model is mainly used for price forecasting. Using information from the obtained price forecasts as the performance measure seems like a natural extension.
- Other ML algorithms can be applied, as discussed in [16].
- The STHTS is generally nonconvex and is often cast as a mixed integer programming (MIP) problem. Reducing the number of constraints in a MIP problem does not necessarily lead to lower computation times. Thus, the framework design needs to be revisited if targeting MIP problems.

References

[1] A. Helseth, M. Haugen, S. Jaehnert, B. Mo, H. Farahmand, and C. Naversen, "Multi-market price forecasting in hydro-thermal power systems," in *Proc. of 15th International Conference on the European Energy Market*, Lodz, Poland, 2018.

[2] A. Haugstad and O. Rismark, "Price forecasting in an open electricity market based on system simulation," in *Proc. of the EPSOM 98 International conference on electrical power systems operation and management*, Zürich, Switzerland, 1998.

[3] C. Weber, *Uncertainty In The Electric Power Industry: Methods And Models For Decision Support*, 1st ed., ser. International Series in Operations Research & Management Science. Springer, 2004.

[4] A. Helseth, G. Warland, and B. Mo, "A Hydrothermal Market Model for Simulation of Area Prices Including Detailed Network Analyses," *International Transactions on Electrical Energy Systems*, vol. 23, no. 8, pp. 1396–1408, 2013.

[5] J. Nowotarski and R. Weron, "Recent advances in electricity price forecasting: A review of probabilistic forecasting," *Renewable and Sustainable Energy Reviews*, vol. 81, pp. 1548–1568, 2018.

[6] K. B. Debnath and M. Mourshed, "Forecasting methods in energy planning models," *Renewable and Sustainable Energy Reviews*, 2018.

[7] F. Ziel and R. Steinert, "Probabilistic mid-and long-term electricity price forecasting," *Renewable and Sustainable Energy Reviews*, vol. 94, pp. 251–266, 2018.

[8] D. vom Stein, N. van Bracht, A. Maaz, and A. Moser, "Development of Adaptive Time Patterns for Multi-Dimensional Power System Simulations," in *Proc. of 14th International Conference on the European Energy Market*, 2017.

[9] G. Savvidis and K. Hufendiek, "Variable time resolution in lp electricity market and investment models," in *Proc. of 15th International Conference on the European Energy Market*, 2018.

[10] S. Pineda and J. M. Morales, "Chronological Time-Period Clustering for Optimal Capacity Expansion Planning With Storage," *IEEE Transactions on Power Systems*, vol. 33, no. 6, pp. 7162–7170, 2018.

[11] A. Turgeon, "Optimal operation of multireservoir power systems with stochastic inflows," *Water Resources Research*, vol. 16, no. 2, pp. 275–283, 1980.

[12] O. Egeland, J. Hegge, E. Kylling, and E. Nes, "The Extended Power Pool Model. Operation Planning of a Multi-River and Multi-Reservoir Hydro-Dominated Power Production System. A Hierarchical Approach," in *CIGRE Paper 32-14*, 1982.

[13] L. A. Terry, M. V. F. Pereira, T. A. A. Neto, L. F. C. A. Silva, and P. R. H. Sales, "Coordinating the Energy Generation of the Brazilian National Hydrothermal Electrical Generating System," *INFORMS Journal on Applied Analytics*, vol. 16, pp. 16–38, 1986.

[14] E. Shayesteh, M. Amelin, and L. Söder, "Multi-Station Equivalents for Short-Term Hydropower Scheduling," *IEEE Transactions on Power Systems*, vol. 31, no. 6, pp. 4616–4625, 2016.

[15] M. Löschenbrand and M. Korpås, "Hydro Power Reservoir Aggregation via Genetic Algorithms," *Energies*, vol. 10, pp. 1–16, 2017.

[16] E. B. Sveen, "Combining mathematical programming and machine learning in electricity price forecasting," M.Sc. Thesis, Norwegian University of Life Sciences, 2019.

[17] M. Haugen, A. Helseth, S. Jaehnert, B. M. annd H. Farahmand, and C. Naversen, "On the importance of detailed thermal modeling for price forecasting in hydro-thermal power systems," in *Proc. of IEEE Canada Electrical Power and Energy Conference (EPEC)*, 2019.

[18] A. Helseth, B. Mo, A. L. Henden, and G. Warland, "Detailed long-term hydro-thermal scheduling for expansion planning in the Nordic power system," *IET Generation, Transmission and Distribution*, vol. 12, no. 2, pp. 441–447, 2018.

[19] L. E. Schäeffer, B. Mo, and I. Graabak, "Electricity Prices and Value of Flexible Generation in Northern Europe in 2030," in *13th International Conference on the European Energy Market*, 2019.

[20] F.-A. Fortin, F.-M. D. Rainville, M.-A. Gardner, M. Parizeau, and C. Gagne, "DEAP: Evolutionary algorithms made easy," *Journal of Machine Learning Research*, vol. 13, no. 2171-2175, 2012.

[21] F. C. et al., "Keras," https://keras.io, 2015.

[22] S. S. Girija, "Tensorow: Large-scale machine learning on heterogeneous distributed systems," Software available from tensorow.org, 2016.

[23] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[24] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural networks for machine learning*, vol. 4, no. 2, pp. 26–31, 2012.

This is the accepted version of an article published in 2020 17th International Conference on the European Energy Market - EEM
http://dx.doi.org/10.1109/EEM49802.2020.9221968