Energy Flexibility Assessment of an Industrial Coldstore Process

J. van der Herten, F. Depuydt, L. De Baets,
D. Deschrijver, M. Strobbe, C. Develder, T. Dhaene
Internet Based Communication Networks and Services (IBCN)
Ghent University - iMinds
Ghent, Belgium.

R. Bruneliere
J-W. Rombouts
REstore NV
Antwerp, Belgium

Abstract—Power-intensive industry plays a key role in balancing supply and demand in the energy grid: by offering flexible power, industry can reduce operating costs and grid operators can avoid technical failures. Recently, research has started to try and address the challenging question of determining the amount of power curtailment (i.e., how much power can be reduced for how long) without violating any process constraints. We consider several machine learning methods to assess the curtailment potential in a coldstore, based on historical data.

I. Introduction

Due to the deregulation of energy markets and the increased production of energy by renewable sources, grid operators worldwide are in need of *flexible* power to balance supply and demand at any time and to avoid technical failures. Although interesting for stability and usability of renewable sources [1], storing power when supply exceeds demand is very costly at this point and therefore economically infeasible. A lot of effort has been put into providing flexibility on the supply side of the grid by introducing versatile solutions such as gas plants that can be enabled quickly when demand rises and might exceed the supply.

Nowadays, regulating the demand side as a function of fluctuations in the power supply is gaining more attention since balancing power consumption can also help to reduce power consumption peaks [2]. A possible example is the regulated and coordinated charging of electric vehicles [3].

Power-intensive industry plays a big role on the demand side, and has the potential to offer flexible power to the grid by optimizing their day-to-day operations. Power supply for processes that have a certain type of buffer capacity can be stopped when there is no immediate negative impact in terms of output, stability or quality of the process. By resuming power supply to these processes when the load on the grid is lower, power consumption is shifted in time to a period with excess capacity, thus avoiding a peak load. This time shift also holds benefits for industry in terms of monetization, due to a lower consumption when power is very expensive.

In order to exploit this flexible power, this paper proposes an innovative implementation for the assessment and modeling of power flexibility of a step-by-step approach which is intellectual property of Restore NV [4]. The novel implementation can be used to assess and quantify energy flexibility of an industrial process at any time and was developed as part of the MonIEFlex ICON research project [5]. It was tested on a real-world coldstore application consisting of several compressors (consuming power) and three different rooms. In Section II, the concept of power flexibility is first formally introduced, and related to earlier work. The collection of the data, and the required processing steps are described in Section III. Section IV explains the application of machine learning to build a predictive model to predict how long power can be curtailed without affecting the process negatively (the time to reach the *boundary condition*). The resulting models are then analysed to identify the impact of the parameters on the predictions in Section V.

II. Power Flexibility

In order to exploit flexible power of industrial processes to improve demand scheduling, the grid operator has to quantify flexible power. Formally this can be expressed as

$$F = \int_0^{\Delta t} \Delta P(t)\, dt \,. \tag{1}$$

This represents the size of the power curtailment $\Delta P > 0$ as a function of time, and the time Δt it is imposed. Assuming ΔP is constant (1) simplifies to

$$F = \Delta P\, \Delta t. \tag{2}$$

The time to reach a critical boundary is affected by internal process values and configuration parameters of the process, as well as external influences such as the weather. In case of the coldstore application, ΔP represents the amount of power saved by disabling the cooling installation. This quantity is known and can be directly observed and modified as it is part of the process control. Obtaining Δt is more complex, as it

is influenced by the state of the process and environmental parameters. Certain state parameters of the process, such as temperature, should not exceed certain *boundary conditions* if ΔP power is curtailed for a time interval: we require that the power curtailment does not impact the process in a negative way (loss of quality, losses, etc.). To determine the total flexiblity, Δt represents the time to reach the boundary condition and should not be exceeded.

In [6], an approach is described to predict the temperature evolution of domestic fridges. Cooling cycles are predicting by matching cycles observed earlier. The temperature of a domestic fridge is defined mostly by the working of its thermostat: the evolution is very periodical and only interrupted by opening doors, which typically occurs only sporadically. An industrial cooling system however results in more complex interactions, making the approach of [6] unreliable under this setting.

A different approach to obtain Δt, is to model all physical relationships with Computational Fluid Dynamics (CFD) simulations with thermal components, using for instance the FloTHERM package [7]. Although these evaluations can be computationally expensive, this method can accurately model the evolution of the temperature. The downside of this approach is it requires a full study of the physical relationships of the process, which makes the solution very specific. Furthermore, it is not always fully known a priori what environmental parameters influence the process: for instance during the MonieFlex project, it was found that for the coldstore case Δt was much more sensitive to the weather than was initially anticipated, even by domain experts.

In this paper an alternative approach is proposed: historical data of the coldstore is collected, analysed and processed to generate a training set for machine learning methods that learn the relationship between its inputs (the state, decision and environmental parameters) and Δt (the output). The resulting predictive model can be used to assess instantaneous flexibility and has the additional benefit that sensitivity analysis can be performed to study the impact of the changes in the inputs to the output. Although this paper focusses on coldstores, this approach is general and can also be applied to different types of industrial processes with minor modifications, i.e., adapt the processing step slightly to compute Δt from the historical data to generate the training sets for the machine learning step.

III. Data Collection and Processing

A schematic overview of the coldstore is given in Fig. 1. Several compressors (consuming $450\,\mathrm{A}$, $370\,\mathrm{A}$ and $200\,\mathrm{A}$ respectively) fill an ammonia buffer. This buffer is consumed by an evaporator in each of the three coldstores (DV01, DV02, DV04). Two coldstores are physically adjacent (DV01 and DV02) whereas DV04 is located in a separate building, as illustrated in Fig. 2.

The compressors can either be on or off, whereas the evaporators can also work at half power. The buffer capacity is small: when the evaporators are running without the compressors enabled, all cooling buffer capacity is consumed within two minutes. On the other hand, enabling a single compressor
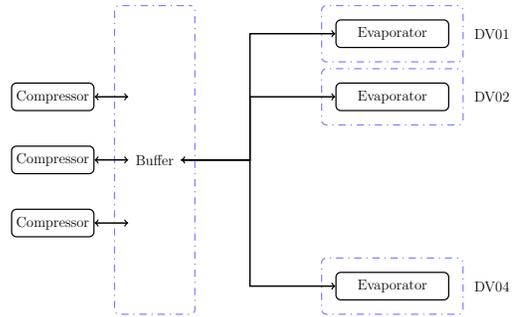

Fig. 1. Schematic illustration of the cooling installation for the coldstore.

Fig. 2. Satellite image of the industrial site, indicating the three coldstore building.

TABLE I
RECORDED PARAMETERS AT THE COLDSTORE SITE

Type	Parameters	Symbols	Unit
State parameters V_s	Coldstore temperatures	$T_{\mathrm{DV01}}, T_{\mathrm{DV02}}$ T_{DV04}	°C
Decision parameters V_d	Compressor states Evaporator states	$C_{\mathrm{A}}, C_{\mathrm{B}}, C_{\mathrm{C}}$ $E_{\mathrm{DV01}}, E_{\mathrm{DV02}}$ E_{DV04}	A on/off
Environmental parameters V_e	Wind speed Wind direction Humidity Outside temperature	W_s W_{dir} H T_{ext}	$\mathrm{m\,s^{-1}}$ ° % °C

without any of the evaporators running builds up the buffer capacity in less than 5 minutes.

Throughout a year, all process-related parameters relevant to estimate the power flexibility were monitored and collected once every minute at the coldstore. An overview is given in Table I. The buffer responds almost instantaneously, so its state can be neglected. The result is a historic set of data represented as a matrix $\tilde{V} \in \mathbb{R}^{N \times d}$. The columns of the matrix are vectors that comprise the evolution of d parameters over time $t = 1, ..., N$.

In order to build a statistical model that predicts Δt, $\tilde{V}$

needs to be processed in two consecutive steps:

1) Obtain the *boundary conditions* $\boldsymbol{V_s}^{\pm}$ for the state parameters $T_{\text{DV01}}, T_{\text{DV02}}, T_{\text{DV01}}$ based on their distribution.
2) For each coldstore x, process $\tilde{\boldsymbol{V}}$ to obtain a dataset consisting of records of the form $(T_x, \boldsymbol{V_d}, \boldsymbol{V_e}, \Delta t)$

The motivation for these steps is given in Section III-A and III-B.

A. Identification of boundary conditions

For the coldstore case, the three temperature parameters for the corresponding coldstores $T_{\text{DV01}}, T_{\text{DV02}}, T_{\text{DV04}}$ are subject to constraints: the coldstores must always remain below a certain threshold, to prevent quality decay of products stored within. For automated derivation of the boundary conditions $\boldsymbol{V_s}^{\pm}$ of the state variables ($\boldsymbol{V_s}$), Self-Organizing Maps (SOM) [8] were applied, using the implementation available in the SOM-Toolbox [9]. The algorithm trains a set of d-dimensional feature vectors (denoted as $\{\boldsymbol{s}_1, ..., \boldsymbol{s}_r\} \subset \mathbb{R}^d$) to fit the topology of $\boldsymbol{V_s}$. The number of nodes r and its structure are automatically determined based on the amount of data vectors and the principal eigenvectors of the supplied data. After training, we choose the bounaries corresponding to the interval defined by the minimum and maximum value of the feature vectors $\boldsymbol{s}$ for that dimension. Fig. 3 illustrates how a SOM is trained to a two-dimensional test dataset, and how the boundaries are derived from the SOM.

Using the described approach, it is found that a critical upper bound for the coldstore temperature in DV01, DV02 and DV04 is $-18\,°\text{C}$. The engineers working at the coldstore factory confirmed this result. Based on tests performed on different data sets, it was found that the approach based on SOMs gave better results on data with a split distribution or correlations compared to simpler approaches such as choosing the 2.5%-97.5% percentile range.

B. Computing time to reach boundary condition Δt

In order to train a machine learning model to predict the time to reach the boundary constraints, this value must first be extracted from the recorded data. Unfortunately, under normal operating conditions the coldstores remain below the boundary conditions of $-18\,°\text{C}$, which means Δt must first be computed. The preprocessing routine described below is applied once to $\tilde{\boldsymbol{V}}$ for all three coldstores. For a coldstore x, all rising segments of the corresponding temperature T are extracted from the time series. The collected segments are further preprocessed to remove the anomalous cases such as missing data, very short rising segments because the evaporators are re-enabled, maintenance etc. A set of data segments is obtained which represent rising temperature curves in the coldstore. According to Newton's law of cooling, the observed temperature values y evolve according to an exponential, so a curve $\tilde{y}$ of the form

$$\tilde{y}(t) = T_0 \exp\left(\frac{\lambda t}{L}\right) \tag{3}$$

is fitted to each segment. T_0 corresponds to the coldstore temperature at the start of the segment, L to the length of the
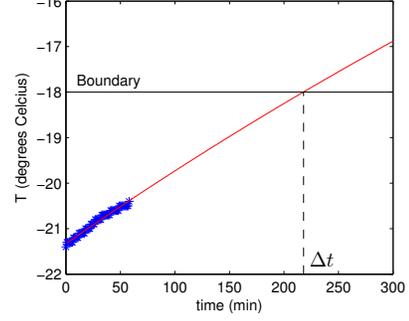

Fig. 4. Illustration of an exponential fit of a rising temperature curve in a coldstore.

segment. The λ parameter is determined through numerical optimization, so the Mean Square Error (MSE) between the curve predictions $\tilde{y}$ and the observed temperatures y is minimized:

$$\text{MSE}(y, \tilde{y}) = \sum_{t=1}^{N} \frac{(y_t - \tilde{y}_t)^2}{N}. \tag{4}$$

By extrapolating the exponential curve, the intersection with the boundary condition yields Δt, expressed in minutes. For this procedure, we assumed the outside temperature (which impacts λ) remains constant. Given the average and maximum values found for Δt in Table II this seems a reasonable assumption. It was pointed out that the temperature of the cargo also influences λ: for the curve fitting this has no specific influence as the optimization determines λ for each curve. However, inclusion of this parameter in $\boldsymbol{V_s}$ would account for this source of variability later on during the modeling process. Unfortunately, the temperature of the cargo was not recorded at the factory, and could not be included.

Each segment is then translated into a data point of the form $(T_x, \boldsymbol{V_d}, \boldsymbol{V_e}, \Delta t)$. Further analysis reveals $\boldsymbol{V_d}$ also consists of irrelevant parameters: only if an evaporator is off, a rising temperature is observed. Because of the properties of the installation this implies the corresponding compressors are also switched off (short buffer). Hence we drop these parameters upfront[1]. Finally $(T_x, \boldsymbol{V_e})$ will be used as model inputs to predict the output Δt. Because DV01 and DV02 are neighbouring coldstores, the temperature of DV02 is also included as additional input parameter for the data points of DV01 and vice-versa, as the temperature of the neighbouring room can potentially influence Δt.

IV. Modeling Δt

In order to build a model of the Δt, the SUMO toolbox [10] was chosen as experimental platform. It contains state-of-the-art machine learning algorithms and automates the overall model building process.

Each dataset representing one the coldstores was randomly split into a training data set (80% of the available entries), and a

---

[1]This can be different if a similar approach is used to compute the flexibility of a different type of process.
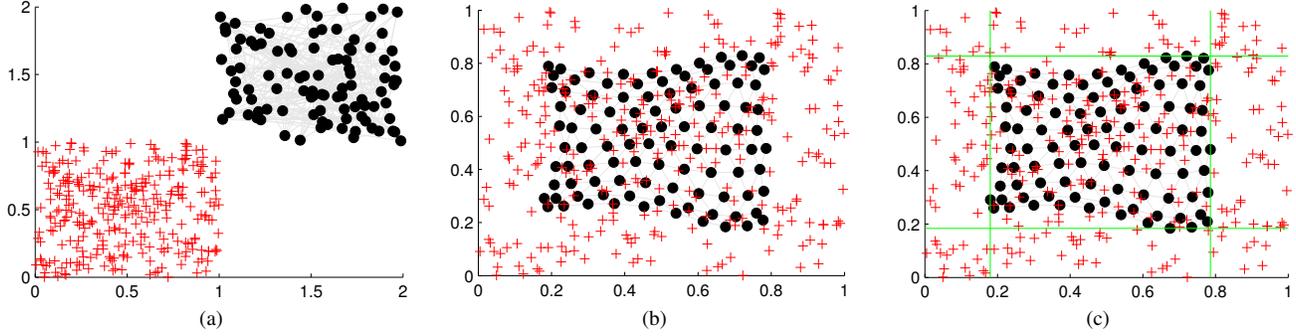
Fig. 3. Illustration of the SOM-based method to define boundary conditions for parameters, on a uniformly generated 2D dataset. Fig. (a) shows the data (red) and the randomly initialized SOM. After training the SOM has adapted to the data topology as shown in (b). The minimum and maximum values of the feature vectors for each parameter are the boundaries as illustrated by the green lines in (c).

TABLE II
DATA CHARACTERISTICS

	DV01	DV02	DV04
Data points	46780	48704	15813
Training set	37424	38963	12650
Testing set	9356	9741	3163
$\Delta t_{\max}$ (min)	290	298	300
$\Delta t_{\min}$ (min)	1.25	1	2
Δt_{avg} (min)	68	57	55
Δt_{med} (min)	52	44	42

separate data set for testing the performance of the final models (20% of the entries). This splitting rate is common: the training set is sufficiently large to perform cross-validation, while the test set is still sufficiently large (see Table II) to ensure it is representative for normal operating conditions. Several models types were trained for each training dataset: Gaussian Processes (GP) [11], [12], Support Vector Regression (SVR) [13], Least-Squares Support Vector Regression (LS-SVR) [14], Extreme Learning Machines (ELM) [15] and Artificial Neural Networks (ANN). Cross-validation using the MSE error function as defined in (4) on the training dataset was chosen as objective function for the hyperparameter optimization step.

The kernel and regularization hyperparameters of SVR and LS-SVR were optimized by applying several global optimization methods (Simulated Annealing [16], Particle Swarm Optimization [17], DiRECT [18] and Pattern Search [19]) and selecting the optimum. For ELM, the number of hidden neurons (up to 5000) and the initialization range of the weights connecting the input layer to the hidden layer were optimized using the same set of global optimizers. To speed up the matrix inversion for large ELM networks GPU acceleration was used. The architecture and the initial weights of the ANN were considered hyperparameters and optimized using a genetic algorithm (10 generations, 10 individuals each). The training function used was Levenberg-Marquard backpropagation with bayesian regularization (300 epochs) [20].

The best model for each model type was then evaluated

on the test set and the Root Mean Square Error (RMSE) was calculated:

$$\mathrm{RMSE}(y, \tilde{y}) = \sqrt{\sum_{t=1}^{N} \frac{(y_t - \tilde{y}_t)^2}{N}}. \tag{5}$$

Furthermore, the Average Absolute Error (AAE):

$$\mathrm{AAE}(y, \tilde{y}) = \frac{1}{N} \sum_{t=1}^{N} |y_t - \tilde{y}_t|, \tag{6}$$

the Average Relative Error (ARE)

$$\mathrm{ARE}(y, \tilde{y}) = \frac{1}{N} \sum_{t=1}^{N} \frac{|y_t - \tilde{y}_t|}{|y_t|}, \tag{7}$$

and the Bayesian Estimated Error Quotient (BEEQ) [21]

$$\mathrm{BEEQ}(y, \tilde{y}) = \left(\prod_{t=1}^{N} \frac{|y_t - \tilde{y}_t|}{|y_t - \bar{y}|} \right)^{\frac{1}{N}} \tag{8}$$

were computed ($\bar{y}$ represents the mean of all values y). Computing this combination of error functions allows an in-depth evaluation of the model performance. The BEEQ error measure is less known, but related to the well known R^2 or Root Relative Square Error [2]. It differs by applying a geometric average instead of an arithmetic average. Like other harmonic error functions (Geometric Average Error or Harmonic Average Error) this makes the BEEQ more positive as it is less affected by extremes (which may occur in our case due to working with real data). For additional information on the properties of the error functions used, the reader is referred to [22].

The results of the modeling experiments are given in Table III. For all three coldstores, there is major difference between the performance of the kernel-based methods, as opposed to the parametric methods. The former methods perform very similar, with only minor differences: for DV01 and DV02 LS-SVR seems to obtain a slightly higher accuracy compared to SVM and GP, for DV04 GP seems to perform very well, yet the AAE

---

[2] $R^2 = 1 - \mathrm{RRSE}^2$

TABLE III
ERRORS FOR THE FINAL MODELS OBTAINED FOR EACH COLDSTORE. THE MSE SCORE IS COMPUTED USING 5-FOLD CROSSVALIDATION AND MINIMIZED
DURING HYPERPARAMETER OPTIMIZATION. ALL OTHER SCORES ARE CALCULATED ON THE TEST SET.

Model type	DV01					DV02					DV04				
	MSE	RMSE	AAE	ARE	BEEQ	MSE	RMSE	AAE	ARE	BEEQ	MSE	RMSE	AAE	ARE	BEEQ
GP	54.5	**3.30**	2.18	**0.08**	0.05	34.4	3.34	2.24	**0.11**	0.07	**63.9**	**4.29**	2.98	**0.17**	0.09
SVR	50.7	3.94	2.27	**0.08**	0.05	33.9	3.46	2.20	0.13	0.06	128	5.30	3.26	0.18	0.09
LS-SVR	**36.2**	3.63	**2.10**	**0.08**	**0.04**	**30.5**	**3.31**	**2.08**	0.12	**0.05**	68.1	4.83	**2.86**	0.18	**0.07**
ELM	1016	31.7	20.7	0.44	0.35	202	9.40	6.07	0.37	0.15	288	13.3	9.62	0.47	0.25
ANN	962	30.8	20.3	0.44	0.34	643	25.1	16.6	0.46	0.36	521	21.5	14.8	0.53	0.36

and BEEQ on the validation set is slightly lower for the LS-SVR model. For DV04, SVM seems to be underperforming about the training data (obtaining a higher crossvalidation score), yet the model performs reasonably on the test set, scoring higher errors mostly for longer Δt values (resulting in a higher AAE while retaining a similar value for ARE). Overall, all three kernel based techniques obtain satisfactory results as indicated by low ARE and BEEQ scores, as well as an AAE of about 2 to 3 minutes.

On the other hand, the scores obtained with the parametric methods are disappointing. Both ELM as well as ANN obtain higher cross-validation scores, which is further reflected in the scores on the validation set. Although still performing better compared to the simplest model available (the mean) as indicated by the BEEQ, the performance is nowhere near to the kernel based results, for all three coldstores. It is possible for different model types to achieve different results, yet the difference between the methods is very significant. A possible reason is a failure of the optimization methods used to find the optimal network architecture, but it occurs for all three coldstores using two different techniques. Further analysis of the models in the next section provides more insight on the response behaviour, providing additional information on the reason why no good architecture is found.

V. SENSITIVITY ANALYSIS

For DV01 and DV02, the LS-SVR models were labeled best, whereas for DV04, the GP model was chosen. Using these models, the Sobol indices [23] were computed to perform sensitivity analysis. This post-processing step provides us information on what parameters (or interactions thereof) influence the output variability the most. Therefore this step assists to understand better what influences Δt most and provides relevant information for power flexibility forecasting. Highly sensitive input parameters must be predicted very accurately in order to minimize the forecasting error.

Table IV displays the first order Sobol indices (main effects, and total effects) for all three coldstores, as well as the residual[3].

Clearly, variable interactions dominate predictions of the best models. This observation could also partially explain the poor performance of the ANN and ELM models: neurons of these networks take linear combinations of input parameters, and pass these through an activation function. In order to approximate the behaviour of interactions, the output of several neurons needs to be aggregated so the interaction is represented as a series expansion. The Sobol indices indicating the main effects are very low (especially for DV01 and DV02), and the residuals indicate a lot of the output variance is approximated by interactions (for the kernel methods). Computation of the higher order indices revealed that this does not stop at second order or third order: given these complex interactions encountered, quite a large network would be required to capture all output variability which also brings a risk of overfitting. Hence, the optimization of the network architecture has failed to find a suitable architecture which provides a good approximation but doesn't overfit the training set. Presenting these interaction terms directly as inputs to the network should increase their performance. A second reason for the performance of the parametric methods could be the absence of the temperature of the cargo when the cooling is stopped. This parameter does affect the time constant of the exponential fits, and introduces a source of variability which is not presented as input parameter.

The total indices reveal that besides the compressor and evaporator variables, all other included variables are important and occur in interactions. The only difference observed is situated between DV01 and DV02 on one hand, and DV04 on the other hand. For the latter, the total indices for the weather parameters are significantly lower, and the temperature of the coldstore itself has a much higher impact (as also indicated by the main effect and the residual of DV04). This is consistent with the fact that the north side of the coldstore is better covered by another building and trees, as observed on Fig. 2.

VI. CONCLUSION

In this paper we introduced an approach developed for prediction of the time to reach a boundary constraint (Δt) in an industrial coldstore process using machine learning, with the goal of computing instantaneous power flexibility. It involves preprocessing historical data to construct a training set, which is then used to learn the relationship between the input parameters and Δt. Application of this approach to a real-world coldstore process indicates Δt can be predicted accurately, but kernel-based methods were better able to capture complex interactions compared to the parametric methods tested in this paper. Further analysis of the final model (for instance using Sobol indices)

---

[3]The percentage of output variance which is caused by higher order interactions.

TABLE IV
FIRST ORDER SOBOL INDICES

	Indices			Total indices		
	DV01	DV02	DV04	DV01	DV02	DV04
T_{DV01}	0.03	0.02	-	0.75	0.50	-
T_{DV02}	< 0.01	0.08	-	0.78	0.63	-
T_{DV04}	-	-	0.34	-	-	0.61
W_s	< 0.01	0.02	0.12	0.67	0.61	0.41
W_{dir}	< 0.01	< 0.01	0.03	0.62	0.57	0.30
H	< 0.01	< 0.01	0.03	0.61	0.56	0.30
T_{ext}	< 0.01	< 0.01	< 0.01	0.51	0.46	0.11
Residual	0.96	0.86	0.47			

can then be used to analyse the influence of the input parameters on Δt which can guide the decision making on how to optimize the process environment as well.

The approach can also be used to predict the power flexibility in the future (*forward flexibility*) which is even more important for load scheduling on the power grid. For the coldstore case, this would mostly involve forecasting of environmental parameters (i.e. weather forecasting). Given the coldstore state, the predicted weather values can be used to predict the flexibility for the following hours or days. The prediction would be as reliable as the weather predictions. This can then further be used for optimization of the power flexibility given the market parameters. These extensions as well as error analysis are subject of further research as part of the MonIEFlex project.

ACKNOWLEDGEMENTS

REFERENCES

[1] V. Virasjoki, P. Rocha, A. S. Siddiqui, and A. Salo, "Market Impacts of Energy Storage in a Transmission-Constrained Power System," 2015.

[2] A.-H. Mohsenian-Rad, V. W. Wong, J. Jatskevich, R. Schober, and A. Leon-Garcia, "Autonomous demand-side management based on game-theoretic energy consumption scheduling for the future smart grid," *Smart Grid, IEEE Transactions on*, vol. 1, no. 3, pp. 320–331, 2010.

[3] K. Mets, T. Verschueren, W. Haerick, C. Develder, and F. De Turck, "Optimizing smart energy control strategies for plug-in hybrid electric vehicle charging," in *Network Operations and Management Symposium Workshops (NOMS Wksps), 2010 IEEE/IFIP*. Ieee, 2010, pp. 293–299.

[4] J.-W. Rombouts, J. Gheerardyn, P.-J. Mermans, and L. Snijers, "Automated demand response energy management system," U.S. Patent 8 417 391, April, 2013.

[5] "iMinds ICON project MonIEFlex," https://www.iminds.be/en/projects/2014/03/20/monieflex, 2015.

[6] V. Lakshmanan, M. Marinelli, A. M. Kosek, F. Sossan, and P. Norgard, "Domestic refrigerators temperature prediction strategy for the evaluation of the expected power consumption," in *Innovative Smart Grid Technologies Europe (ISGT EUROPE), 2013 4th IEEE/PES*. IEEE, 2013, pp. 1–5.

[7] B. Zahn *et al.*, "Evaluating thermal characterization accuracy using CFD codes-a package level benchmark study of IcePak TM and Flotherm R," in *Thermal and Thermomechanical Phenomena in Electronic Systems, 1998. ITHERM'98. The Sixth Intersociety Conference on*. IEEE, 1998, pp. 322–329.

[8] T. Kohonen and S.-O. Maps, "Springer series in information sciences," *Self-organizing maps*, vol. 30, 1995.

[9] J. Vesanto, J. Himberg, E. Alhoniemi, and J. Parhankangas, "Self-organizing map in Matlab: the SOM Toolbox," in *Proceedings of the Matlab DSP conference*, vol. 99, 1999, pp. 16–17.

[10] D. Gorissen, K. Crombecq, I. Couckuyt, P. Demeester, and T. Dhaene, "A Surrogate Modeling and Adaptive Sampling Toolbox for Computer Based Design," *Journal of Machine Learning Research*, vol. 11, pp. 2051–2055, 2010. [Online]. Available: http://sumo.intec.ugent.be

[11] C. E. Rasmussen and H. Nickisch, "Gaussian processes for machine learning (GPML) toolbox," *The Journal of Machine Learning Research*, vol. 11, pp. 3011–3015, 2010.

[12] I. Couckuyt, D. Deschrijver, and T. Dhaene, "Fast calculation of multiobjective probability of improvement and expected improvement criteria for pareto optimization," *Journal of Global Optimization*, pp. 1–20, 2013.

[13] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[14] J. A. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, J. Vandewalle, J. Suykens, and T. Van Gestel, *Least squares support vector machines*. World Scientific, 2002, vol. 4.

[15] G.-B. Huang, D. H. Wang, and Y. Lan, "Extreme learning machines: a survey," *International Journal of Machine Learning and Cybernetics*, vol. 2, no. 2, pp. 107–122, 2011.

[16] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi *et al.*, "Optimization by simulated annealing," *science*, vol. 220, no. 4598, pp. 671–680, 1983.

[17] J. Kennedy, "Particle swarm optimization," in *Encyclopedia of Machine Learning*. Springer, 2010, pp. 760–766.

[18] D. R. Jones, C. D. Perttunen, and B. E. Stuckman, "Lipschitzian optimization without the Lipschitz constant," *Journal of Optimization Theory and Applications*, vol. 79, no. 1, pp. 157–181, 1993.

[19] C. Audet and J. E. Dennis Jr, "Analysis of generalized pattern searches," *SIAM Journal on Optimization*, vol. 13, no. 3, pp. 889–903, 2002.

[20] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the Marquardt algorithm," *IEEE Transactions on Neural Networks*, vol. 5, no. 6, pp. 989–993, 1994.

[21] X. R. Li and Z. Zhao, "Evaluation of estimation algorithms part I: incomprehensive measures of performance," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 42, no. 4, pp. 1340–1358, 2006.

[22] D. Gorissen, I. Couckuyt, E. Laermans, and T. Dhaene, "Multiobjective global surrogate modeling, dealing with the 5-percent problem," *Engineering with Computers*, vol. 26, no. 1, pp. 81–98, 2010.

[23] I. M. Sobol, "Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates," *Mathematics and computers in simulation*, vol. 55, no. 1, pp. 271–280, 2001.