Proceedings of the 18th Mediterranean Electrotechnical Conference
MELECON 2016, Limassol, Cyprus, 18-20 April 2016

Adaptive Modeling and Sampling Methodologies for Internet of Things Applications

Joachim van der Herten, Ivo Couckuyt, Dirk Deschrijver, Piet Demeester and Tom Dhaene
Internet Based Communication Networks and Services (IBCN)
Ghent University - iMinds
Gaston Crommenlaan 8 (Bus 201), B-9050 Gent, Belgium
Email: joachim.vanderherten@intec.ugent.be

***Abstract*—The past few years several cloud services offer automated machine learning software. This enables non-experts to build sophisticated predictive models so they can focus on their area of expertise instead, and use these state-of-the-art machine learning techniques. These were the same principles that guided the development of the surrogate modeling (SUMO) toolbox to assist engineers during (virtual) product design and rapid prototyping with state-of-the-art machine learning methods. A proof of concept was developed, which exposes the technologies of the SUMO toolbox as a network service, offering them to the devices attached to the same network. Both the implementation of the service as well as the possibilities for Internet of Things are discussed.**

I. Introduction

Direct experiments in the physical world are often infeasible or impractical for many science and engineering design applications (e.g., car crashes, earthquake propagation, etc.). Therefore often complex physics-based simulation code is used to perform virtual experiments. The downside of using these simulation codes is their heavy computational demands. A single evaluation of high-fidelity simulations may require hours, days or even weeks.

To expedite this process, the construction of a cheap-to-evaluate approximation constructed on a small set of simulator evaluations which mimics the simulator response has become a standard approach. The data-driven approximation (known as surrogate model, metamodel or response surface model) is then used for optimization, sensitivity analysis, design space exploration, or any other application relying heavily on evaluation. Several approximation methods have been successfully used in many applications over the years [1]–[5].

For this approach to be effective, only a small amount of data points evaluated by the simulator (*samples*) should suffice to build an accurate approximation. Traditionally the data points were all chosen upfront, followed by a modeling phase with automated optimization of the model parameters (*hyperparameter optimization*). This approach often results in either too much data points evaluated (oversampled), or a lack of evaluations which results in an inaccurate model (undersampled). To improve this workflow it was turned into an iterative process (referred to as sequential design) with an *adaptive sampling* algorithm selecting additional data points in each iteration as needed, guided by all information that has already been retrieved in earlier iterations.

Earlier several methodologies for sequential design, such as automated modeling and optimal selection of new data points, have been bundled into the SUMO Toolbox [6], a state-of-the-art tool for surrogate modeling with sequential design. The toolbox hides a lot of complexity and automates many tasks, allowing its users (primarily engineers) to focus on their goals (e.g., product design) instead. This focus makes the toolbox also suitable for other applications if these methodologies were available as a network service, also referred to as Machine Learning-as-a-Service (MLaaS). This can be particularly useful in the context of Internet of Things (IoT): the user of the toolbox would no longer be a physical person, but a different network service or an attached device requiring for instance accurate predictions, assistance with measurements or optimization for its internal process.

This article first describes the SUMO Toolbox and its design philosophy in Section II. In Section III the principles of MLaaS and the existing technologies are explained. Furthermore, several use cases are given where methodologies of the SUMO toolbox can assist devices and applications in IoT. Finally, the implementation prototype is discussed in Section IV.

II. SUMO Toolbox

The SUMO Toolbox was designed primarily as research platform for fully automated adaptive construction of surrogate models featuring high extensibility. Given a data set or simulator, the aim is constructing a surrogate model automatically given many possible approaches: problems are different in nature and will typically require a dedicated solution strategy. The entire process is automated as model parameters are automatically tuned, and samples can be selected on the fly.

By default, the toolbox is configured according to the principles of sequential design as illustrated in Figure 1. The process initiates from a set of data points, which are evaluated (using simulation codes in the context of surrogate modeling). An intermediate model is built and its *hyperparameters* are chosen automatically. When the model is sufficiently accurate (or any other stopping condition has been met) the final model is returned. If this is not the case an adaptive sampling algorithm picks additional data points for evaluation and the process repeats by simulation of these points. When the intermediate models are not required by the adaptive sampling routine, it is also possible to run the adaptive sampling

978-1-5090-0058-6/16/$31.00 ©2016 IEEE

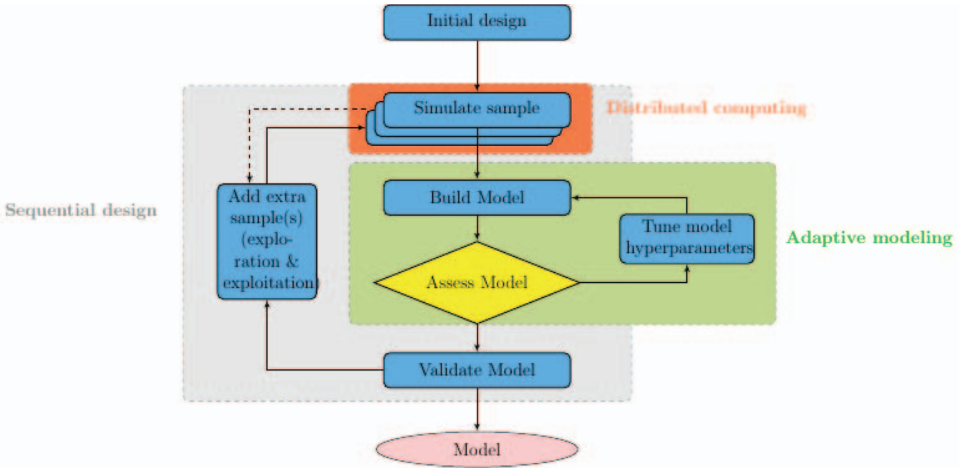

Fig. 1. Flowchart of surrogate modeling with sequential design. An initial data set is simulated and automatically modeled. When the model does not meet the requirements, an adaptive sampling algorithm selects additional data points, which are simulated and another model is built. This continues until the preset modeling goals have been met, or any other stopping criterion is met.

routine a few times without model building (as indicated by the dashed arrow). The toolbox has been used previously in several successful engineering applications in several domains including electronics [7].

The toolbox configuration is represented by means of an XML file, allowing the configuration and composition of all different plugins into the modeling workflow for a specific problem. Many different plugins are available for the different sub-problems: model types include rational functions, Kriging, [8], splines, Support Vector Machines (SVM) [9]–[11], Artificial Neural Networks (ANN), Extreme Learning Machines (ELM) [12], Least Squares-SVM (LS-SVM) [13] or Random Forests [14]. Hyperparameter optimization algorithms include Particle Swarm Optimization [15], Efficient Global Optimization [16] (commonly referred to as Bayesian optimization), simulated annealing, and Genetic Algorithms. Model selection can be done with cross-validation, but also AIC, a Leave-out set and LRM [17] are available, as well as sample selection approaches such as random, error based, density based [18], [19] or hybrid methods [20]. Popular Design of Experiments methodologies for computer experiments such as Latin hypercube [21], [22] or Box-Bhenken are included as well. Evaluation of samples can be done locally or distributed on a cluster when expensive simulation code is used.

III. Machine Learning-as-a-Service (MLaaS)

A. Description

Recently, several big companies including Amazon [23], Google [24], IBM [25], Microsoft [26] or bigML [27] have deployed cloud services for automated machine learning. This is often referred to as Machine Learning-as-a-service (MLaaS) following the cloud paradigms SaaS, PaaS and IaaS (although MLaaS is not a new paradigm, depending on what is offered it is covered by at least one of these three). The main purpose of these services is to offer a fast and scalable access to Machine Learning technology to model data and deploy a prediction service. Roughly, the functionality of these services can be characterized as follows:

- Assembling a clean data set from several sources (flat text, RDBMS, NoSQL stores, ...). Functionality is offered to remove highly correlated variables, handle missing values, apply mathematical transforms, etc.
- Automated modeling: machine learning algorithms for supervised (regression or classification) and unsupervised learning can be applied to the data set to build a predictive model. These models are completely autonomous and automatically tune their parameters. The optimal model type can be chosen automatically as well.
- Quality assessment of the model using test data to verify its performance.
- Automated deployment of the model as a cloud service. This allows other applications to use the model, using an API.

The research area that aims to automate these tasks is referred to as AutoML [28]: the primary objective is user-friendliness and automated approaches requiring no input other than the data allowing users without a machine learning background to build predictions. The original design goals of the SUMO toolbox are very closely related to these principles as the motivation was to develop a tool for automated modeling and sequential data collection, allowing engineers to focus on product design instead of machine learning [29], [30]. This implies many of the modeling and sampling methodologies in the SUMO toolbox were implemented with these principles in mind (especially automated modeling and quality assessment).

Following these developments, effort was made to improve accessibility to the implementations of the SUMO toolbox (i.e., exposing the functionality over the network). Our primary goal is the integration of adaptive modeling and sampling methodologies in *Internet of Things* (IoT) applications: in this setting the SUMO toolbox will permit devices to focus on their specific task instead of machine learning, as it already does for the design engineer.

B. Applications for IoT

The availability of SUMO toolbox components as a network service (implementation is discussed in Section IV) facilitates usage for automated machine learning in IoT applications. Prime candidates are the implemented models and automated model building strategies: several algorithms for hyperparameter optimization including an approach which also selects the optimal model type [31], [32]. Possible use cases include applications and devices submitting data to the service. When the model is ready it can be used for prediction purposes such as prediction of temperature, signal throughput, output quality of a production process etc. This is similar to what is offered by the corporate cloud providers, although the final model is not hosted as a cloud service but transmitted as a mathematical expression. This offers specific advantages for usage of the service by IoT devices which are not necessarily permanently connected to the network: the model is evaluated locally (on the embedded system) and can still be used when the device is no longer connected. In addition, the SUMO Toolbox can be instructed to prefer compact models during model training (through multi-objective hyperparameter optimization [33]): this is very useful to generate small models for devices with only limited computational power (at the expense of model accuracy).

A second class of toolbox components that are relevant for application of machine learning in IoT are the sampling algorithms. In sequential design, these approaches collect the data optimally to improve the model towards a specific goal with a minimum amount of evaluations. When the goal is finding an optimum, Efficient Global Optimization (EGO) [16] is typically used: this algorithm constructs a Gaussian Process from the evaluated data points, and uses its prediction variance to calculate the Expected Improvement (EI) criterion to pick the next evaluation. In literature, this is often referred to as Bayesian optimization [34]. This approach can be used by device(s) that require process calibration (for instance the parameters of a communication protocol between several sensors, with the objective of maximizing throughput): run information is submitted to the service and when the job is started, it will iteratively suggest a new configuration and ask to report the performance on the objective. As EGO is known to require only a limited number of iterations, a good configuration can be found within acceptable time and the additional cost of the network communication is minimal. Furthermore, other (multi-objective) surrogate-based optimization techniques such as Efficient Multi-objective Optimization are supported as well [35]. A potential application of this functionality is on-the-fly optimization of parameters of network communication protocols, in order to maximize the throughput [5].

When the goal is an accurate model, space-filling sequential methods [18] can be used, or methods that analyse the responses [20], [36] and intermediate models [37] to increase sampling density in areas that are more difficult to model (poles, discontinuities, strong non-linearities etc). The benefit of the latter is illustrated in Figure 2: instead of purely space-filling, the generated experimental design increases focus on a non-linear region. This is specifically useful for devices which are used to actively acquire data, such as for instance drone-based measuring [38], [39], or devices such as robot vacuum cleaners. Rather than performing simulations, a drone collects real-time measurements. The common goal remains to acquire as much information as possible given a limited time span (as batteries run out). The drone contacts the service for the optimal route it has to fly, and submits those measurements to retrieve the next path. At the same time, the service can already apply online model building, to assure models are ready when the measurement process is finalized.

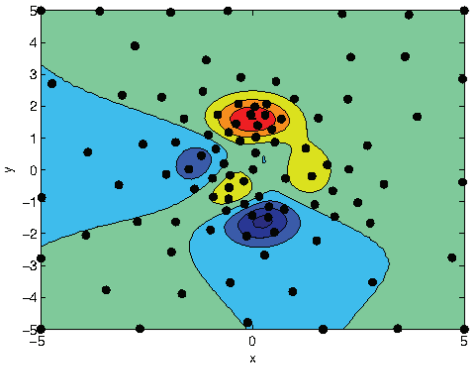

Fig. 2. Peaks function sampled with the FLOLA-Voronoi [36] method for sequential design. Clearly, the algorithm has focussed on the central regions which is more dynamic compared to the flat surroundings.

IV. SUMO-MLaaS Implementation

The SUMO toolbox has been implemented in MATLAB and has grown into a mature tool for product design over the years. The release of MATLAB R2015a introduces a Python library, offering significant interaction possibilities. This greatly accelerated the development of a network-based service functionality for the SUMO toolbox. A prototype of a multi-threaded service has been implemented in Python: an illustration of the architecture is given in Figure 3. The SUMO-MLaaS service on the master node controls one or several MATLAB processes with the SUMO toolbox installed and loaded, and serves as an endpoint towards devices. Jobs can be submitted and started through an XML-RPC interface (using the native Python library), which also offers the functionality to submit data files. It is also possible to run the service configured in slave mode on different nodes to allow running several jobs concurrently across multiple machines. Locally, the parallel mode of the SUMO Toolbox can be enabled to take advantage of multi-core CPU architectures.

To expose an interface to monitor jobs and display their results, an integrated webservice built on top of Tornado [40] was developed. It offers the content of the output directory of a run (including the model, profiler information, model plots, the toolbox configuration used for the run and various other information regarding the run) over the HTTP protocol. The information is also aggregated into a webpage. Secondly Tornado also sets up a WebSocket to collect and process the log messages of the SUMO instances. As the SUMO Toolbox has always used the native Java Logging API, this only required adding a specific handler which sends the messages to the WebSocket. Note that the service automatically generates the required configuration for the logger and adds it to the SUMO configuration before a run is started.

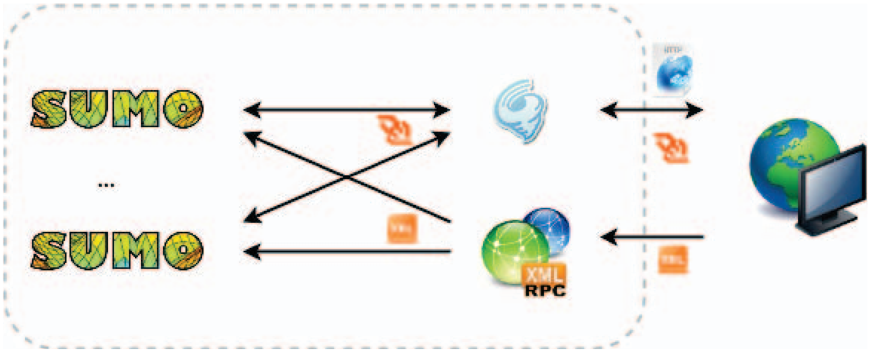

Fig. 3. Overview of the SUMO-MLaaS architecture. SUMO instances are controlled by a network service, exposing an XML-RPC interface to submit jobs. IoT devices (including lightweight) can utilize the functionality, and interact with the runs through a tornado webserver.

Tornado also sets up a public WebSocket interface. When connecting to this WebSocket it is possible to request the logging messages of a running job on one of the SUMO instances: this allows development of all forms of (possibly automated) monitoring applications to keep IoT devices informed of the modeling progress. Furthermore it can also be used to send responses when a SUMO run with sequential design requests a new data point (for instance when SUMO is used for Bayesian optimization). The SUMO toolbox collects this data internally by a specific datasource plugin which listens a second internal WebSocket offered by Tornado when data has been requested. Sharing the first internal WebSocket was impossible due to the microkernel architecture of the toolbox, with the functionality implemented in two different plugins.

Once a model has been trained and validated, exporting the model can be done by downloading either the SUMO model which can be used on a local MATLAB installation, or by obtaining an automatically generated function (containing a mathematical expression) in javascript or Python. The function encodes the mathematical expression which corresponds to the SUMO model, and can be loaded into applications or devices attached to the network. This feature can be used for fast and automated integration of the predictive models.

V. Conclusion

As a mature toolbox for surrogate modeling used during product design, the SUMO Toolbox offers many automated modeling and sampling methodologies which allow engineers worldwide to focus on product design instead of tuning machine learning models. Exposing these advanced techniques to the network as a machine learning service opens up many possibilities, especially for IoT devices and applications. They can relay their machine learning efforts, and obtain predictions when the modeling has finished. Furthermore, the sampling algorithms for optimization can be used to optimize processes in IoT (both on a single device as well as a group of devices cooperating). Sampling algorithms developed for global surrogate modeling may assist devices that are actively acquiring data such as drones to maximize the amount of information that can be extracted with a limited amount of measurements

The automated modeling techniques share similarities with other AutoML initiatives. We do not claim the methods implemented in the SUMO Toolbox outperform for instance those of the commercial cloud providers. Rather they can be used as a non-commercial alternative. Furthermore, the solutions offered by cloud providers come with user friendly interfaces while our current focus is offering automated machine learning to IoT devices, although our architecture allows development of such interface later on. As an additional advantage, surrogate models are essentially computationally cheap mathematical expressions which can be transferred to the devices for local evaluation: this is very suitable for usage in an IoT context. The sampling techniques target a more specific range of IoT applications. To our knowledge, these kind of algorithms are not (yet) readily available off the shelf in other services.

Acknowledgment

This research has (partially) been funded by the Interuniversity Attraction Poles Programme BESTCOM initiated by the Belgian Science Policy Office. Ivo Couckuyt is a postdoctoral research fellow of FWO-Vlaanderen.

References

[1] F. Declercq, I. Couckuyt, H. Rogier, and T. Dhaene, “Complex permittivity characterization of textile materials by means of surrogate modelling,” in *Antennas and Propagation Society International Symposium (APSURSI), 2010 IEEE*. IEEE, 2010, pp. 1–4.

[2] K. Goethals, I. Couckuyt, T. Dhaene, and A. Janssens, “Sensitivity of night cooling performance to room/system design: Surrogate models based on CFD,” *Building and Environment*, vol. 58, pp. 23–36, 2012.

[3] D. Deschrijver, F. Vanhee, D. Pissoort, and T. Dhaene, “Automated Near-Field Scanning Algorithm for the EMC Analysis of Electronic Devices,” *IEEE Transactions on Electromagnetic Compatibility*, vol. 54, no. 3, pp. 502–510, 2012.

[4] S. Koziel, A. Bekasiewicz, I. Couckuyt, and T. Dhaene, “Efficient multi-objective simulation-driven antenna design using co-kriging,” *Antennas and Propagation, IEEE Transactions on*, vol. 62, no. 11, pp. 5900–5905, 2014.

[5] M. T. Mehari, E. De Poorter, I. Couckuyt, D. Deschrijver, J. Vanhie-Van Gerwen, D. Pareit, T. Dhaene, and I. Moerman, “Efficient global optimization of multi-parameter network problems on wireless testbeds,” *Ad Hoc Networks*, vol. 29, pp. 15–31, 2015.

[6] 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

[7] D. Gorissen, D. Deschrijver, T. Dhaene, and D. D. Zutter, “A software framework for automated behavioral modeling of electronic devices [application notes],” *Microwave Magazine, IEEE*, vol. 13, no. 6, pp. 102–118, 2012.

[8] I. Couckuyt, T. Dhaene, and P. Demeester, “ooDACE Toolbox: A Flexible Object-Oriented Kriging Implementation,” *Journal of Machine Learning Research*, vol. 15, pp. 3183–3186, 2014.

[9] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the fifth annual workshop on Computational learning theory*. ACM, 1992, pp. 144–152.

[10] 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.

[11] J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in *Advances in large margin classifiers*. Citeseer, 1999.

[12] 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.

[13] 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.

[14] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

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

[16] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient Global Optimization of Expensive Black-Box Functions," *J. of Global Optimization*, vol. 13, no. 4, pp. 455–492, 1998.

[17] D. Gorissen, "Grid-enabled adaptive surrugate modeling for computer aided engineering," Ph.D. dissertation, Ghent University, 2010.

[18] K. Crombecq, I. Couckuyt, D. Gorissen, and T. Dhaene, "Space-filling Sequential Design Strategies for Adaptive Surrogate Modelling," in *Soft Computing Technology in Civil, Structural and Environmental Engineering (CSC 2009)*, vol. 92, 2009, pp. Paper–50.

[19] K. Crombecq, E. Laermans, and T. Dhaene, "Efficient space-filling and non-collapsing sequential design strategies for simulation-based modeling," *European Journal of Operational Research*, vol. 214, no. 3, pp. 683–696, 2011.

[20] K. Crombecq, D. Gorissen, D. Deschrijver, and T. Dhaene, "A Novel Hybrid Sequential Design Strategy for Global Surrogate Modelling of Computer Experiments," *SIAM Journal of Scientific Computing*, vol. 33, no. 4, pp. 1948–1974, 2010.

[21] E. Dam, B. van Husslage, D. den Hertog, and J. Melissen, "Maximin Latin hypercube designs in two dimensions," *Operations Research*, vol. 55, no. 1, pp. 158–169, 2007.

[22] F. A. Viana, G. Venter, and V. Balabanov, "An algorithm for fast optimal Latin hypercube design of experiments," *International journal for numerical methods in engineering*, vol. 82, no. 2, pp. 135–156, 2010.

[23] Amazon Inc., "Amazon Web Services: Predictive Analytics," 2015. [Online]. Available: https://aws.amazon.com/machine-learning

[24] Google Inc., "Prediction API," 2010. [Online]. Available: https://cloud.google.com/prediction

[25] P. Zikopoulos, K. Parasuraman, T. Deutsch, J. Giles, D. Corrigan *et al.*, *Harness the power of big data The IBM big data platform*. McGraw Hill Professional, 2012.

[26] T. Redkar and T. Guidici, *Windows Azure Platform*. Springer, 2009.

[27] F. J. Martin, O. Rovira, J. Verwoerd, P. Petersen, C. Parker, J. A. Ortega, B. Garcia, J. J. DONALDSON, A. Blasco, A. Ashenfelter *et al.*, "Predictive modeling and data analysis in a secure shared system," Sep. 12 2013, uS Patent App. 14/025,063.

[28] F. Hutter, B. Kégl, R. Caruana, I. Guyon, H. Larochelle, and E. Viegas, "Automatic Machine Learning (AutoML)," in *ICML 2015 Workshop on Resource-Efficient Machine Learning, 32nd International Conference on Machine Learning*, 2015.

[29] D. Gorissen, K. Crombecq, W. Hendrickx, and T. Dhaene, "Grid enabled metamodeling," in *In Proc. of 7th International Meeting on High Performance Computing for Computational Science (VECPAR 2006)*, 2006.

[30] D. Gorissen, K. Crombecq, I. Couckuyt, and T. Dhaene, "Automatic approximation of expensive functions with active learning," in *Foundations of Computational, Intelligence Volume 1*. Springer, 2009, pp. 35–62.

[31] D. Gorissen, L. De Tommasi, J. Croon, and T. Dhaene, "Automatic Model Type Selection with Heterogeneous Evolution: An application to RF circuit block modeling," in *Proceedings of the IEEE Congress on Evolutionary Computation, WCCI 2008, Hong Kong*, 2008.

[32] I. Couckuyt, D. Gorissen, F. D. Turck, and T. Dhaene, "Automatic surrogate model type selection during the optimization of expensive black-box problems," in *Winter Simulation Conference*, 2011, pp. 4269–4279.

[33] D. Gorissen, I. Couckuyt, E. Laermans, and T. Dhaene, "Pareto-based multi-output metamodeling with active learning," in *Proceedings of the 11th International Conference on Engineering Applications of Neural Networks (EANN 2009), London, England*, 2009.

[34] R. Martinez-Cantin, "BayesOpt: a Bayesian optimization library for nonlinear optimization, experimental design and bandits," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 3735–3739, 2014.

[35] 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*, vol. 60, no. 3, pp. 575–594, 2014.

[36] J. van der Herten, I. Couckuyt, D. Deschrijver, and T. Dhaene, "A Fuzzy Hybrid Sequential Design Strategy for Global Surrogate Modeling of High-Dimensional Computer Experiments," *SIAM Journal on Scientific Computing*, vol. 37, no. 2, pp. A1020–A1039, 2015.

[37] W. Hendrickx and T. Dhaene, "Sequential design and rational metamodelling," in *Proceedings of the 2005 Winter Simulation Conference*, M. Kuhl, S. N. M., F. B. Armstrong, and J. A. Joines, Eds., Dec. 2005, pp. 290–298.

[38] G. Giebel, U. Schmidt Paulsen, J. Bange, A. La Cour-Harbo, J. Reuder, S. Mayer, A. van der Kroonenberg, and J. Mølgaard, "Autonomous Aerial Sensors for Wind Power Meteorology-A Pre-Project," Danmarks Tekniske Universitet, Risø Nationallaboratoriet for Bæredygtig Energi, Tech. Rep., 2012.

[39] B. Subramanian, N. Chokani, and R. S. Abhari, "Drone-Based Experimental Investigation of Three-Dimensional Flow Structure of a Multi-Megawatt Wind Turbine in Complex Terrain," *Journal of Solar Energy Engineering*, vol. 137, no. 5, p. 051007, 2015.

[40] M. Dory, A. Parrish, and B. Berg, *Introduction to Tornado*. O'Reilly Media, Inc., 2012.