Combining Machine Learning and Simulation to a Hybrid Modelling Approach: Current and Future Directions

Laura von Rueden[1,2(✉)], Sebastian Mayer[1,3], Rafet Sifa[1,2], Christian Bauckhage[1,2], and Jochen Garcke[1,3,4]

[1] Fraunhofer Center for Machine Learning, Sankt Augustin, Germany
[2] Fraunhofer IAIS, Sankt Augustin, Germany
[3] Fraunhofer SCAI, Sankt Augustin, Germany
[4] Institute for Numerical Simulation, University of Bonn, Bonn, Germany
laura.von.rueden@iais.fraunhofer.de

Abstract. In this paper, we describe the combination of machine learning and simulation towards a hybrid modelling approach. Such a combination of data-based and knowledge-based modelling is motivated by applications that are partly based on causal relationships, while other effects result from hidden dependencies that are represented in huge amounts of data. Our aim is to bridge the knowledge gap between the two individual communities from machine learning and simulation to promote the development of hybrid systems. We present a conceptual framework that helps to identify potential combined approaches and employ it to give a structured overview of different types of combinations using exemplary approaches of simulation-assisted machine learning and machine-learning assisted simulation. We also discuss an advanced pairing in the context of Industry 4.0 where we see particular further potential for hybrid systems.

Keywords: Machine learning · Simulation · Hybrid approaches

1 Introduction

Machine learning and *simulation* have a similar goal: To predict the behaviour of a system with data analysis and mathematical modelling. On the one side, machine learning has shown great successes in fields like image classification [21], language processing [24], or socio-economic analysis [7], where causal relationships are often only sparsely given but huge amounts of data are available. On the other side, simulation is traditionally rooted in natural sciences and engineering, e.g. in computational fluid dynamics [35], where the derivation of causal relationships plays an important role, or in structural mechanics for the performance evaluation of structures regarding reactions, stresses, and displacements [6].

However, some applications can benefit from combining machine learning and simulation. Such an hybrid approach can be useful when the processing

M. R. Berthold et al. (Eds.): IDA 2020, LNCS 12080, pp. 548–560, 2020.
https://doi.org/10.1007/978-3-030-44584-3_43

capabilities of classical simulation computations can not handle the available dimensionality of the data, for example in earth system sciences [30], or when the behaviour of a system that is supposed to be predicted is based on both known, causal relationships and unknown, hidden dependencies, for example in risk management [25]. However, such challenges are in practice often still approached distinctly with either machine learning or simulation, apparently because they historically originate from distinct fields. This raises the question how these two modelling approaches can be combined into a hybrid approach in order to foster intelligent data analysis. Here, a key challenge in developing a hybrid modelling approach is to bridge the knowledge gap between the two individual communities, which are mostly either experts for machine learning or experts for simulation. Both groups have extremely deep knowledge about the methods used in their particular fields. However, the respectively used terminologies are different, so that an exchange of ideas between both communities can be impeded.

Related work that describes a combination of machine learning with simulation can roughly be divided in two groups, not surprisingly, either from a machine learning or a simulation point of view. The first group frequently describes the integration of simulation into machine learning as an additional source for training data, for example in autonomous driving [23], thermodynamics [19], or biomedicine [13]. A typical motivation is the augmentation of data for scenarios that are not sufficiently represented in the available data. The second group of related works describes the integration of machine learning techniques in simulation, often for a specific application, such as car crash simulation [6], fluid simulation [38], or molecular simulation [26]. A typical motivation is to identify surrogate models [16], which offer an approximate but cheaper to evaluate model to replace the full simulation. Another technique that is used to adapt a dynamical simulation model to new measurements is data assimilation, which is traditionally used in weather forecasting [22]. Related work that considers an equal combination of machine learning and simulation is quite rare. A work that is closest to describing such a hybrid, symbiotic modelling approach is [4].

More general, the integration of prior knowledge into machine learning can be described as *informed machine learning* [34] or *theory-guided data science* [18]. The paper [34] presents a survey with a taxonomy that structures approaches according to the knowledge type, representation, and integration stage. We reuse those categories in this paper. However, that survey considers a much broader spectrum of knowledge representations, from logic rules over simulation results to human interaction, while this paper puts an explicit focus on simulations.

Our goal is to make the key components of the two modelling approaches *machine learning* and *simulation* transparent and to show the versatile, potential combination possibilities in order to inspire and foster future developments of hybrid systems. We do not intend to go into technical details but rather give a high-level methodological overview. With our paper we want to outline a vision of a stronger, more automated interplay between data- and simulation-based analysis methods. We mainly aim our findings at the data analysis and machine

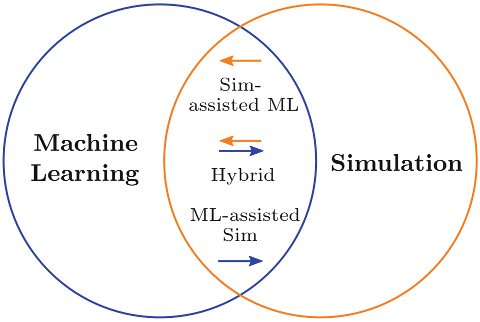

Fig. 1. Subfields of Combining Machine Learning and Simulation. The fields of machine learning and simulation have an intersecting area, which we partition into three subfields: 1. Simulation-assisted machine learning describes the integration of simulations into machine learning. 2. Machine-learning assisted simulation describes the integration of machine learning into simulation. 3. A hybrid combination describes a combination of machine learning and simulation with a strong mutual interplay.

learning community, but also those from the simulation community are welcome to read on. Generally, our target audience are researchers and users of one of the two modelling approaches who want to learn how they can use the other one.

The contributions of this paper are: 1. A conceptual framework serving as an orientation aid for comparing and combining machine learning and simulation, 2. a structured overview of combinations of both modelling approaches, 3. our vision of a hybrid approach with a stronger interplay of data- and simulation based analysis.

The paper is structured as follows: In Sect. 2 we give a brief overview of the subfields that result from combining machine learning and simulation. In Sect. 3 we present these two separate modelling approaches along our conceptual framework. In Sect. 4 we describe the versatile combinations by giving exemplary references and applications. In Sect. 5 we further discuss our observations in Industry 4.0 projects that lead us to a vision for the advanced pairing of machine learning and simulation. Finally we conclude in Sect. 6.

2 Overview

In this section, we give a short overview about the subfields that result from a combination of machine learning with simulation. We view the combination with equal focus on both fields, driving our vision of a hybrid modelling approach with a stronger and automated interplay. Figure 1 illustrates our view on the fields' overlap, which can be partitioned into the three subfields simulation-assisted machine learning, machine-learning assisted simulation, and a hybrid combination. Even though the first two can be regarded as one-sided approaches because they describe the integration with a point of view from one approach, the last one can be regarded as a two-sided approach. Although the term *hybrid*

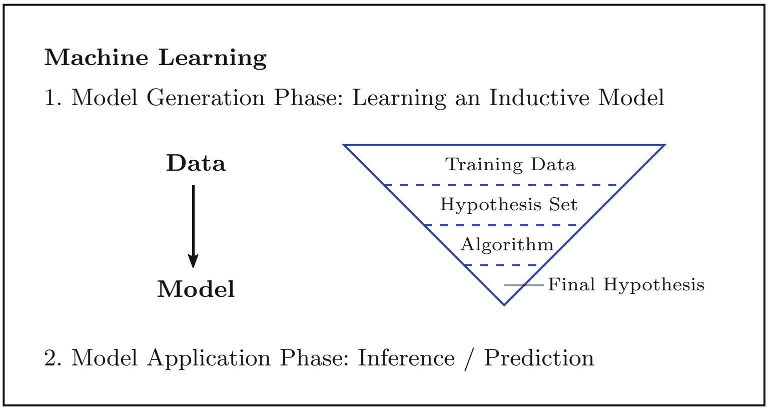
Machine Learning
1. Model Generation Phase: Learning an Inductive Model
Data
Model
Training Data
Hypothesis Set
Algorithm
Final Hypothesis
2. Model Application Phase: Inference / Prediction

Fig. 2. Components of Machine Learning. Machine Learning consists of two phases 1. model generation, and 2. model application, where the focus is usually made on the first phase, in which an inductive model is learned from data. The components of this phase are the training data, a hypothesis set, a learning algorithm, and a final hypothesis [1,34]. It describes the finding of patterns in an initially large data space, which are finally represented in a condensed form by the final hypothesis. This is illustrated by the reversed triangle and can be described as a "bottom-up approach".

is in the literature often used for the above one-sided approaches, we prefer to use it only for the two-sided approach where machine learning and simulation have a strong mutual, symbiotic-like interplay.

3 Modelling Approaches

In this section, we describe the two modelling approaches by means of a conceptual framework that aims to make them and their components transparent and comparable.

3.1 Machine Learning

The main goal of machine learning is that a machine automatically learns a model that describes patterns in given data. The typical components of machine learning are illustrated in Fig. 2. In the first, main phase an inductive model is learned. Inductive means that the model is built by drawing conclusions from samples and is thus not guaranteed to depict causal relationships, but can instead identify hidden, previously unknown patterns, meaning that the model is usually not knowledge-based but rather data-based. This inductive model can finally be applied to new data in order to predict or infer a desired target variable.

The model generation phase can be roughly split into four sub-phases or respective components [1,34]. Firstly, training data is prepared that depicts historical records of the investigated process or system. Secondly, a hypothesis set

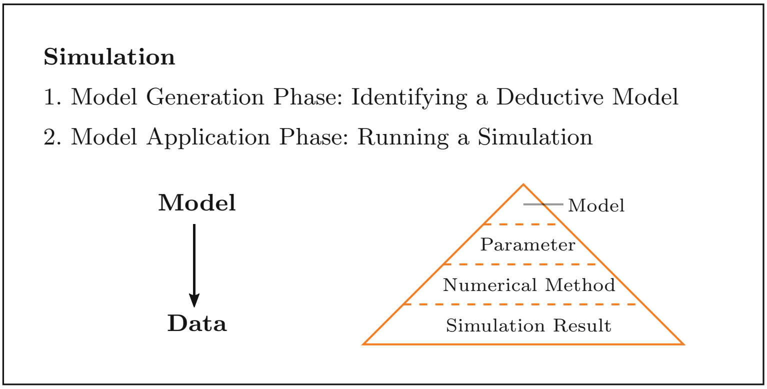
Simulation
1. Model Generation Phase: Identifying a Deductive Model
2. Model Application Phase: Running a Simulation
Model
Data
Model
Parameter
Numerical Method
Simulation Result

Fig. 3. Components of Simulation. Simulation comprises the two phases 1. model generation, and 2. model application, where the focus often is on the second phase, in which an earlier identified deductive model is used in order to create simulation results. The components of this phase are the simulation model, input parameters, a numerical method, and the simulation result. It describes the unfolding of local interactions from a compactly represented initial model into an expanded data space. This is supposed be illustrated by the triangle and can be described as a "top-down approach".

is defined in the form of a function class or network architecture that is assumed to map input features to the target variables. Thirdly, a learning algorithm tunes the parameters of the hypothesis set so that the performance of the mapping is maximized by using optimization algorithms like gradient descent and results in, fourthly, the final hypothesis, which is the desired inductive model. This model generation phase is often repeated in a loop-like manner by tuning hyper-parameters until a sufficient model performance is achieved.

3.2 Simulation

The goal of a simulation is to predict the behaviour of a system or process for a particular situation. There are different types of simulations, ranging from cellular automata, over agent-based simulations, to equation-based simulations [9,15,36]. In the following we concentrate on the last type, which is based on mathematical models and is especially used in science and engineering. The first, required stage preceding the actual simulation is the identification of a deductive model, often in the form of differential equations. Deductive in this context means that the model describes causal relationships and can thus be called knowledge-based. Such models are often developed through extensive research, starting with a derivation, for example in theoretical physics, and continuing with plentiful experimental validations. Some recent research exists of proof-of-concepts for identifying models directly from data [8,33].

The main phase of a simulation is the application of the identified model for a specific scenario, often called running a simulation. This phase can be described in four typical main components or sub-phases, which are, as illustrated in Fig. 3, the mathematical model, the input parameters, the numerical

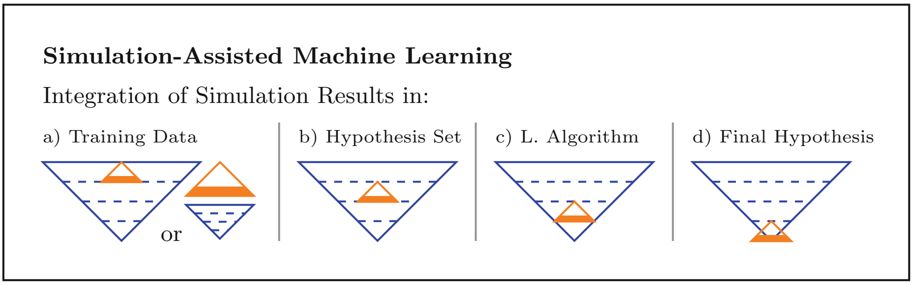
Simulation-Assisted Machine Learning
Integration of Simulation Results in:
a) Training Data
or
b) Hypothesis Set
c) L. Algorithm
d) Final Hypothesis

Fig. 4. Types of Simulation-Assisted Machine Learning. Simulations, in particular the simulation results, can be generally integrated into the four different components of machine learning. The triangles illustrate the machine learning (blue/dark gray) or the simulation (orange/light gray) approach and their components, which are themselves presented in Figs. 2 and 3. The simulation results can be used to (a) augment the training data, (b) define parts of the hypothesis set in the form of empirical functions, (c) steer the training algorithm in generative adversarial networks, or (d) verify the final hypothesis against scientific consistency. (Color figure online)

method, and finally the simulation result [36]. After the selection of a mathematical model, the input parameters that describe the specific scenario are defined in the second sub-phase. They can comprise general parameters such as the spatial domain or time of interest, as well as initial conditions quantifying the systems' or processes' initial status and boundary conditions defining the behaviour at domain borders. In the third sub-phase, a numerical method computes the solution of the given model observing the constraints resulting from the input parameters. Examples for numerical methods are finite differences, finite elements or finite volume methods for spatial discretization [36], or particle methods based on interaction forces [26]. These form the basis for an approximate solution, which is the final simulation result. This model application phase is often repeated in a loop-like manner, e.g., by tuning the discretization to achieve a desired approximation accuracy and stability of the solution.

4 Combining Machine Learning and Simulation

In this section, we describe combinations of machine learning and simulation by using our conceptual framework from Sect. 3. Here, we focus on simulation-assisted machine learning and machine-learning assisted simulation. For each of the methodical combination types, we give exemplary application references.

4.1 Simulation-Assisted Machine Learning

Simulation offers an additional source of information for machine learning that goes beyond typically available data and that is rich of knowledge. This additional information can be integrated into the four components of machine learning as illustrated in Fig. 4. In the following, we will give an overview about these integration types by giving for each an illustrative example and refer for a more detailed discussion to [34].

Simulations are particularly useful for creating additional training data in a controlled environment. This is for example applied in autonomous driving, where simulations such as physics engines are employed to create photo-realistic traffic scenes, which can be used as synthetic training data for learning tasks like semantic segmentation [14], or for adversarial test generation [40]. As another example, in systems biology, simulations can be integrated in the training data of kernelized machine learning methods [13].

Moreover, simulations can be integrated into the hypothesis set, either directly as the solvers or through deduced, empirical functions that compactly describe the simulations results. These functions can be built into the architecture of a neural network, as shown for the application of finding an optimal design strategy for a warm forming process [20].

The integration of simulations into the learning algorithm can for example be realized by generative adversarial networks (GANs), which learn a prediction function that obeys constraints, which might be unknown but are implicitly given through a simulation [31].

Another important integration type is in the validation of the final hypothesis by simulations. An example for this comes from material discovery, where first a machine learning model suggests new compounds based on patterns in a data basis, and second the physical properties are computed and thus checked by a density functional theory simulation [17].

An approach that uses simulations along the whole machine learning pipeline is reinforcement learning (RL), when the model is learned in a simulated environment [2]. Studies under the keyword "sim-to-real" are often concerned with robots learning to grip or move unknown objects in simulations and usually require retraining in reality. An application for controlling the temperature of plasma follows the analogous approach, i.e., a training based on a software-physics model, where the learned RL model is then further adapted for use in reality [41].

4.2 Machine-Learning Assisted Simulation

Machine learning is often used in simulation with the intention to support the solution process or to detect patterns in the simulation data. With respect to our conceptual framework presented in Sect. 3, machine learning techniques can be used for the initial model, the input parameters, the numerical method, and the final simulation results, as illustrated in Fig. 4. In the following we will give an overview about the integration types. Again, we do not intend to cover the full spectrum of machine-learning assisted simulation, we rather want to illustrate its diverse approaches through representative examples.

A prominent integration type of machine learning techniques into simulation is the identification of simpler models, such as surrogate models [11,12,16,26].

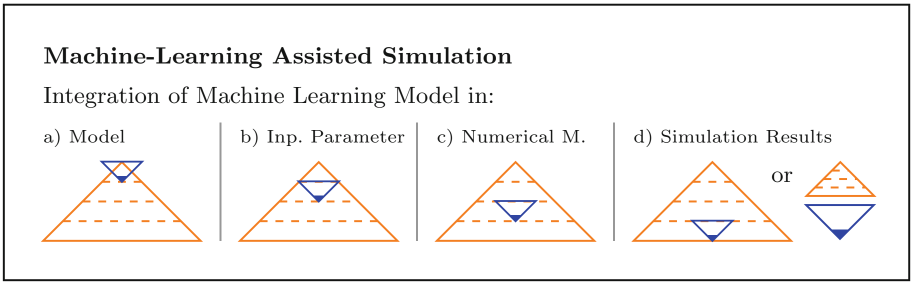
Machine-Learning Assisted Simulation
Integration of Machine Learning Model in:
a) Model
b) Inp. Parameter
c) Numerical M.
d) Simulation Results
or

Fig. 5. Types of Machine-Learning Assisted Simulation. Machine learning techniques, in particular the final hypothesis, can be used in different simulation components. The triangles illustrate the machine learning (blue/dark gray) or the simulation (orange/light gray) approach and their components, which are themselves explained in Figs. 2 and 3. Exemplary use cases for machine learning models in simulation are (a) model order reduction and the development of surrogate models that offer approximate but simpler solutions, (b) the automated inference of an intelligent choice of input parameters for a next simulation run, (c) a partly trainable solver for differential equations, or d) the identification of patterns in simulation results for scientific discovery. (Color figure online)

These are approximate and cheap to evaluate models that are particularly of interest when the solution of the original, more precise model is very time- or resource-consuming. The surrogate model can then be used to analyse the overall behaviour of the system in order to reveal scenarios that should be further investigated with the detailed original simulation model. Such surrogate models can be developed with machine-learning techniques either with data from real-world experiments, or with data from high-fidelity simulations. One application example is the optimization of process parameters using deep neural networks as surrogate models [27]. Kernel-based approaches are also commonly used as surrogate models for simulations, an example to improve the energetic efficiency of a gas transport network is shown in [10]. A well-established approach for surrogate modelling is model order reduction, for example with proper orthogonal decomposition, which is closely related to principal component analysis [5,37].

Data assimilation, which includes the calibration of constitutive models and the estimation of system states, is another area where machine learning techniques enhance simulations. Data assimilation problems can be modelled using dynamic Bayesian networks with continuous physically interpretable state spaces where the evaluation of transition kernels and observation operators requires forward-simulation runs [29].

Machine learning techniques can also be used to study the parameter dependence of simulation results. For example, after an engineer executes a sequence of simulations, a machine learning model can detect different behavioral modes in the results and thus reduce the analysis effort during the engineering process [6]. This supports the selection of the parameter setting for the next simulation, for which active learning techniques can also be employed. For example, [39] studied it for selecting the molecules for which the internal energy shall be determined by computationally expensive quantum-mechanical calculations, as well as for determining a surrogate model for the fluid flow in a well-bore while drilling.

The integration of machine learning techniques into the numerical method can support to obtain the numerical solution. One approach is to exchange parts of the model that are resource-consuming to solve, with learned models that can be computed faster, for example with machine learning generated force fields in molecular dynamics simulations [26]. Another approach that is recently investigated are trainable solvers for partial differential equations that determine the complete solution through a neural network [28].

A further, very important integration type is the application of machine learning techniques on the simulation results in order to detect patterns, often motivated by the goal of scientific discovery. While there are plenty of application domains, two exemplary representatives are particle physics [3] and earth-sciences, for example with the use of convolutional neural networks for the detection of weather patterns on climate simulation data [30]. For further examples we refer to a survey about explainable machine learning for scientific discovery [32].

5 Advanced Pairing of Machine Learning and Simulation

Section 4 gave a brief overview of the versatile existing approaches that integrate aspects of machine learning into simulation and vice versa, or that combine simulation and machine learning sequentially. Yet, we think that the integration of these two established worlds is only at the beginning, both in terms of modelling approaches and in terms of available software solutions.

In the following, we describe a number of observations from our project experience in the development of cyber-physical systems for Industry 4.0 applications that support this assessment. Note that the key technical goal of Industry 4.0 is the flexibilization of production processes. In addition to the broad integration of digital equipment in the production machinery, a key provider of flexibilization is a decrease of process design and dimensioning times and ideally, a merging of planning and production phase that are today still strictly separated. This requires a new generation of computer-aided engineering (CAE) software systems that allow for very fast process optimization cycles with real time feedback loops to the production machinery. An advanced pairing of machine learning and simulation will be key to realize such systems by addressing the following issues:

- **Simulation results are not fully exploited:** Especially in the industrial practice, simulations are run with a very specific analysis goal based on expert-designed quantities of interest. This ignores that the simulation result might reveal more patterns and regularities, which might be irrelevant for the current analysis goal but useful in other contexts.
- **Selective surrogate modelling**: Even if modern machine learning approaches are used, surrogate models are built for very specific purposes and the decision when and where to use a surrogate model is left to domain

experts. In this way, it is exploited too little that similar underlying systems might lead to similar surrogate models and in consequence, too many costly high-fidelity simulations are run to generate the data basis, although parts of the learned surrogate models could be transferred.

- **Parameter studies and simulation engines:** Parameter and design studies are well-established tools in many fields of engineering. Surprisingly, the frameworks to conduct these studies and to build the surrogate models are third-party solutions that are separated from the core simulation engines. For the parameter study framework, the simulation engine is a black box, which does not know that it is currently used for a parameter study. In turn, the standard rules to generate sampling points in the parameter space are not aware about the internals of the simulation engine. This raises the question how much more efficient parameter studies could be conducted so that both software systems were stronger connected to each other.

These observations lead us to a research concept that we propose in this paper and call it **learning simulation engines**. A learning simulation engine is a hybrid system that combines machine learning and simulation in an optimal way. Such an engine can automatically decide when and where to apply learned surrogate models or high-fidelity simulations. Surrogate models are efficiently organized and re-used through the use of transfer learning. Parameter and design optimization is an integral component of the learning simulation engine and active learning methods allow the efficient re-use of costly high-fidelity computations.

Of course, the vision of a learning simulation engine raises numerous research questions. We describe some of them in view of Fig. 1. First of all, the question is how learning and simulation can be technically combined to such an advanced hybrid approach, especially, if they can only be integrated into each other by using the final simulation results and the final hypothesis (as shown in Figs. 4 and 5), or if they can also be combined at an earlier sub-phase. Moreover, the counterparts of the learning's model generation phase and the simulation's model application phase (see Figs. 2 and 3) should be investigated further in order to better understand the similarities and differences to the simulation's model generation phase and a learning's model application phase.

6 Conclusion

In this paper, we described the combination of machine learning and simulation motivated by fostering intelligent analysis of applications that can benefit from a combination of data- and knowledge-based solution approaches.

We categorized the overlap between the two fields into three sub-fields, namely, simulation-assisted machine learning, machine-learning assisted simulation, and a hybrid approach with a strong and mutual interplay. We presented a conceptual framework for the two separate approaches, in order to make them and their components transparent for the development of a potential combined approach. In summary, it describes machine learning as a bottom-up approach that generates an inductive, data-based model and simulation as a top-down approach that applies a deductive, knowledge-based model. Using this conceptual framework as an orientation aid for their integration into each other, we gave a structured overview about the combination of machine learning and simulation. We showed the versatility of the approaches through exemplary methods and use cases, ranging from simulation-based data augmentation and scientific consistency checking of machine learning models, to surrogate modelling and pattern detection in simulations for scientific discovery. Finally, we described the scenario of an advanced pairing of machine learning and simulation in the context of Industry 4.0 where we see particular further potential for hybrid systems.

References

1. Abu-Mostafa, Y.S., Magdon-Ismail, M., Lin, H.T.: Learning From Data (2012)
2. Akkaya, I., et al.: Solving rubik's cube with a robot hand (2019). arXiv:1910.07113
3. Albertsson, K., Altoe, P., Anderson, D., Andrews, M., Espinosa, J.P.A., Aurisano, A., Basara, L., Bevan, A., Bhimji, W., et al.: Machine learning in high energy physics community white paper. J. of Phys.: Conf. Ser. **1085**, 022008 (2018)
4. Baker, R.E., Pena, J.M., Jayamohan, J., Jérusalem, A.: Mechanistic models versus machine learning, a fight worth fighting for the biological community? Biol. Lett. **14**(5), 20170660 (2018)
5. Benner, P., Gugercin, S., Willcox, K.: A survey of projection-based model reduction methods for parametric dynamical systems. SIAM Rev. **57**(4), 483–531 (2015)
6. Bohn, B., Garcke, J., Iza-Teran, R., Paprotny, A., Peherstorfer, B., Schepsmeier, U., Thole, C.A.: Analysis of car crash simulation data with nonlinear machine learning methods. Proc. Comput. Sci. **18**, 621–630 (2013)
7. Bollen, J., Mao, H., Pepe, A.: Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. In: AAAI Conference Weblogs and Social Media (2011)
8. Brunton, S.L., Proctor, J.L., Kutz, J.N.: Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proc. Nat. Acad. Sci. **113**(15), 3932–3937 (2016)
9. Bungartz, H.J., Zimmer, S., Buchholz, M., Pflger, D.: Modeling and Simulation (2014)
10. Clees, T., Hornung, N., Nikitin, I., Nikitina, L., Steffes-lai, D.: RBF-metamodel driven multi-objective optimization and its applications. Int. J. Adv. Intell. Syst. **9**(1), 19–24 (2016)
11. Cozad, A., Sahinidis, N.V., Miller, D.C.: Learning surrogate models for simulation-based optimization. AIChE J. **60**(6), 2211–2227 (2014)
12. Cranmer, K., Brehmer, J., Louppe, G.: The frontier of simulation-based inference (2019). arXiv:1911.01429
13. Deist, T.M., Patti, A., Wang, Z., Krane, D., Sorenson, T., Craft, D.: Simulation-assisted machine learning. Bioinformatics **35**(20), 4072–4080 (2019)
14. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: Carla: An open urban driving simulator (2017). arXiv:1711.03938
15. Durán, J.M.: Computer Simulations in Science and Engineering. TFC. Springer, Heidelberg (2018). https://doi.org/10.1007/978-3-319-90882-3

16. Forrester, A., Sobester, A., Keane, A.: Engineering Design via Surrogate Modelling: A Practical Guide. John Wiley, Hoboken (2008)
17. Hautier, G., Fischer, C.C., Jain, A., Mueller, T., Ceder, G.: Finding natures missing ternary oxide compounds using machine learning and density functional theory. Chem. Mater. **22**(12), 3762–3767 (2010)
18. Karpatne, A., Atluri, G., Faghmous, J.H., Steinbach, M., Banerjee, A., Ganguly, A., Shekhar, S., Samatova, N., Kumar, V.: Theory-guided data science: a new paradigm for scientific discovery from data. IEEE Trans. Knowl. Data Eng. **29**(10), 2318–2331 (2017)
19. Karpatne, A., Watkins, W., Read, J., Kumar, V.: Physics-guided neural networks (pgnn): an application in lake temperature modeling (2017). arXiv:1710.11431
20. Kim, H.S., Koc, M., Ni, J.: A hybrid multi-fidelity approach to the optimal design of warm forming processes using a knowledge-based artificial neural network. Int. J. Mach. Tools Manuf. **47**(2), 211–222 (2007)
21. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS (2012)
22. Lahoz, W., Khattatov, B., Menard, R. (eds.): Data Assimilation. Making Sense of Observations. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-540-74703-1
23. Lee, K.H., Li, J., Gaidon, A., Ros, G.: Spigan: Privileged adversarial learning from simulation. In: ICLR (2019)
24. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: NIPS (2013)
25. Mitchell-Wallace, K., Foote, M., Hillier, J., Jones, M.: Natural Catastrophe Risk Management and Modelling: A practitioner's Guide. John Wiley, Hoboken (2017)
26. Noé, F., Tkatchenko, A., Müller, K.R., Clementi, C.: Machine learning for molecular simulation (2019). arXiv:1911.02792
27. Pfrommer, J., Zimmerling, C., Liu, J., Kärger, L., Henning, F., Beyerer, J.: Optimisation of manufacturing process parameters using deep neural networks as surrogate models. Proc. CIRP **72**(1), 426–431 (2018)
28. Raissi, M., Perdikaris, P., Karniadakis, G.E.: Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations (2017). arXiv:1711.10561
29. Reich, S., Cotter, C.: Probabilistic Forecasting and Bayesian Data Assimilation. Cambridge University Press, Cambridge (2015)
30. Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., et al.: Deep learning and process understanding for data-driven earth system science. Nature **566**(7743), 195–204 (2019)
31. Ren, H., Stewart, R., Song, J., Kuleshov, V., Ermon, S.: Adversarial constraint learning for structured prediction. In: IJCAI (2018)
32. Roscher, R., Bohn, B., Duarte, M.F., Garcke, J.: Explainable machine learning for scientific insights and discoveries (2020). IEEE Access
33. Rudy, S.H., Brunton, S.L., Proctor, J.L., Kutz, J.N.: Data-driven discovery of partial differential equations. Sci. Adv. **3**(4), e1602614 (2017)
34. von Rueden, L., Mayer, S., Beckh, K., Georgiev, B., Giesselbach, S., Heese, R., Kirsch, B., Pfrommer, J., Pick, A., Ramamurthy, R., Walczak, M., Garcke, J., Bauckhage, C., Schuecker, J.: Informed machine learning - a taxonomy and survey of integrating knowledge into learning systems (2020). arXiv:1903.12394v2
35. Shaw, C.T.: Using Computational Fluid Dynamics (1992)
36. Strang, G.: Computational Science and Engineering, vol. 791 (2007)

37. Swischuk, R., Mainini, L., Peherstorfer, B., Willcox, K.: Projection-based model reduction: formulations for physics-based machine learning. Comput. Fluids **179**, 704–717 (2019)
38. Tompson, J., Schlachter, K., Sprechmann, P., Perlin, K.: Accelerating Eulerian fluid simulation with convolutional networks. In: ICML (2017)
39. Tsymbalov, E., Makarychev, S., Shapeev, A., Panov, M.: Deeper connections between neural networks and gaussian processes speed-up active learning (2019). arXiv:1902.10350
40. Tuncali, C.E., Fainekos, G., Ito, H., Kapinski, J.: Simulation-based adversarial test generation for autonomous vehicles with machine learning components. In: IEEE Intelligent Vehicles Symposium (2018)
41. Witman, M., Gidon, D., Graves, D.B., Smit, B., Mesbah, A.: Sim-to-real transfer reinforcement learning for control of thermal effects of an atmospheric pressure plasma jet plasma sources. Sci. Technol. **28**(9), 095019 (2019)