

Driven by Data or Derived Through Physics? A Review of Hybrid Physics Guided Machine Learning Techniques With Cyber-Physical System (CPS) Focus

RAHUL RAI¹ AND **CHANDAN K. SAHU**

Manufacturing and Design Lab (MADLab), Department of Mechanical and Aerospace Engineering, University at Buffalo, Buffalo, NY 14260, USA

Corresponding author: Rahul Rai (rahulrai@buffalo.edu)

This work supported in part by the Defense Advanced Research Projects Agency (DARPA) under Agreement HR0011-18-9-0037, and in part by the Naval Surface Warfare Center (NSWC) NEEC under Award N00174-19-0025.

ABSTRACT A multitude of cyber-physical system (CPS) applications, including design, control, diagnosis, prognostics, and a host of other problems, are predicated on the assumption of model availability. There are mainly two approaches to modeling: Physics/Equation based modeling (Model-Based, MB) and Machine Learning (ML). Recently, there is a growing consensus that ML methodologies relying on data need to be coupled with prior scientific knowledge (or physics, MB) for modeling CPS. We refer to the paradigm that combines MB approaches with ML as hybrid learning methods. Hybrid modeling (HB) methods is a growing field within both the ML and scientific communities, and are recognized as an important emerging but nascent area of research. Recently, several works have attempted to merge MB and ML models for the complete exploitation of their combined potential. However, the research literature is scattered and unorganized. So, we make a meticulous and systematic attempt at organizing and standardizing the methods of combining ML and MB models. In addition to that, we outline five metrics for the comprehensive evaluation of hybrid models. Finally, we conclude by shedding some light on the challenges of hybrid models, which we, as a research community, should focus on for harnessing the full potential of hybrid models. An additional feature of this survey is that the hybrid modeling work has been discussed with a focus on modeling cyber-physical systems.

INDEX TERMS Cyber-physical systems, deep learning, deep neural networks, hybrid models, model-based, machine learning, physics guided, physics informed, physics prior, theory guided.

I. INTRODUCTION

Just as the **microscope** empowered our naked eyes to see cells, microbes, and viruses, thereby advancing the progress of biology and medicine; or as the **telescope** opened our minds to the immensity of the cosmos and has enabled humankind to explore and understand the space in spectacular detail, in the current century, **ComputingScope** (computational “instruments” for “viewing” and analyzing data) will help successfully decipher and navigate another infinite: the staggeringly complex multi-modal (text, image, video, and sensors) information in all facets of our lives.

The associate editor coordinating the review of this manuscript and approving it for publication was Fan-Hsun Tseng².

ComputingScope will give us a vision of the whole and help us synthesize actionable information. Recent success of Deep Learning (DL) [1]–[8] acts as a harbinger of the potential of ComputingScope. While the DL has made a significant impact in domains such as natural language processing (NLP), computer vision, and speech recognition, its impact on cyber-physical systems (CPS) is only recently being understood. In order to realize the potential of DL frameworks in CPS applications, there is a growing consensus that ML methodologies relying on data need to be coupled with physics-based modeling (prior knowledge) techniques.

The technology push driving the integration of a myriad variety of sensors, computational intelligence, communication, and control to bring “smartness” into a variety of CPSs

is the immediate next frontier for ComputingScopes. CPSs are complex systems capable of Cognition, Communication, Computation and Control (4C) [9]–[13]. Consequently, CPS has four main components with unique functions. They are physical infrastructure consisting of sensors for sensing the stimuli to curate raw data, actuators for responding to the stimuli, a computational intelligence platform for computing the response contingent upon the stimuli, and a network that interconnects all of them. Figure 1 illustrates the architecture and functioning of a CPS. The development of intelligent *computational techniques* (*computation component of the CPS*) that are capable of harnessing CPS sensor curated raw data to generate actionable intelligence (typically hidden in data) is the focus of this review.

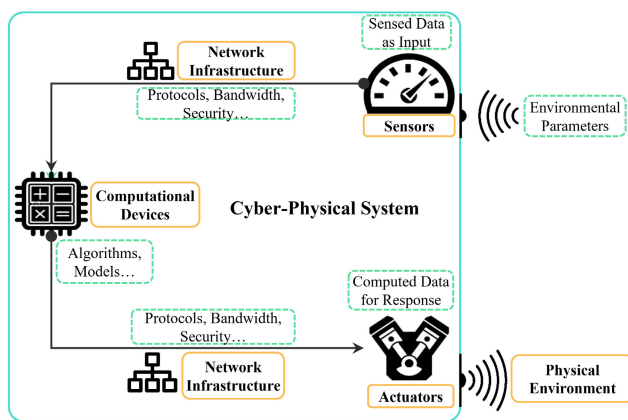


FIGURE 1. Schematic diagram of a Cyber-Physical System depicting the manner in which the physical (Sensors, actuators, computational devices, network infrastructure) and cyber (Sensed and computed data, algorithms, system models, network protocols, bandwidth etc.) components are intertwined.

A multitude of CPS applications, including design, control, diagnosis, prognostics, and a host of other problems, are predicated on the assumption of model availability. There are mainly two approaches to modeling: Physics/Equation based modeling (Model-Based, MB) and Machine Learning (ML). Model-based methods assume the availability of an accurate system model, while data-driven methods are based on machine learning. Purely data-driven ML methods ignore any knowledge about the physical/abstract systems. Additionally, ML approaches require a large amount of labeled training data that is typically unavailable. MB approaches require excellent physics models and good specification of parameter values. When building models of complex CPSs, we are often limited by the unavailability of the parameters of the system components due to incomplete technical specifications, hidden physical interactions, or interactions that are too complex to model from first principles. Hence, we often make simplifying assumptions (e.g., linear approximations) and construct coarse models that imperfectly describe the behavior of the real system. A prudent approach is to use hybrid methods that use the physics of the system and prior knowledge about the domain to guide the construction of machine learning

techniques such as Deep Neural Networks (DNNs). The principal goal of this review is to discuss challenges related to the development of hybrid methods that combine multi-physics equation-based models with data-driven machine learning models (such as DNNs) to enable predictive modeling of CPSs in the presence of imperfect models and sparse and noisy data.

Our review is complementary to the existing review in the hybrid modeling domain [14]. The key differences include but are not limited to (1) our focus is different and is primarily on CPS applications, (2) we outline a completely different framework for categorizing hybrid modeling techniques, (3) we provide an update on recent hybrid modeling papers in the last five years with a primary focus on deep-learning techniques, and (4) we outline several new directions for research in CPS hybrid modeling domain.

The paper is organized into eight sections. Section II discusses the existing techniques of modeling CPS. The hurdles faced by the existing CPS modeling techniques are elaborated in section III. The necessity of combining ML and MB for eliminating those hurdles are outlined in section IV. The methods for combining ML and MB models are discussed in section V. The metrics for evaluating the performance of the hybrid models are deliberated in section VI. The new challenges which may emanate from hybrid modeling are explored in section VII. And finally, section VIII concludes the paper. Figure 2 summarizes the main components of our work.

II. MODELING CYBER-PHYSICAL SYSTEMS

A cyber-physical system is an embodiment of physical entities and accompanying processes with computational capabilities to link the physical world of sensors and actuators with the virtual world of information processing [15]–[20]. The term ‘Cyber-Physical System’ was coined by Helen Gill from NSF in 2006 [21]. The federal Networking and Information Technology Research and Development (NITRD) [22] envisions that CPSs can revolutionize agriculture [23]–[26], building control [27]–[30], security [31]–[33], energy management [34], hazard response [35]–[38], healthcare [16], [39], [40], society [41], [42], transportation [43], [44], environment [45] and manufacturing industries [46]–[49]. Even after exploded research interests and investments, the research in CPSs is still in a nascent stage. Detailed review outlining the-state-of-the-art in CPS domain can be found in [16], [31], [50]–[53].

Progress in CPS automation can be seen as belonging to three eras. Era one, where all the *dirty dangerous* [54] work involving onerous human labor, industrial equipment was automated. This was succeeded by automation of the *dull* [54] activities, which consisted of automated interfaces, clerical chores, service transactions, etc. And finally, now, in the third era, CPS is being empowered to make *decisions* [54]. In the third era, AI in general and machine learning, in particular, is going to assist CPS in making automated decisions to accomplish physical tasks. Previously, automation captured

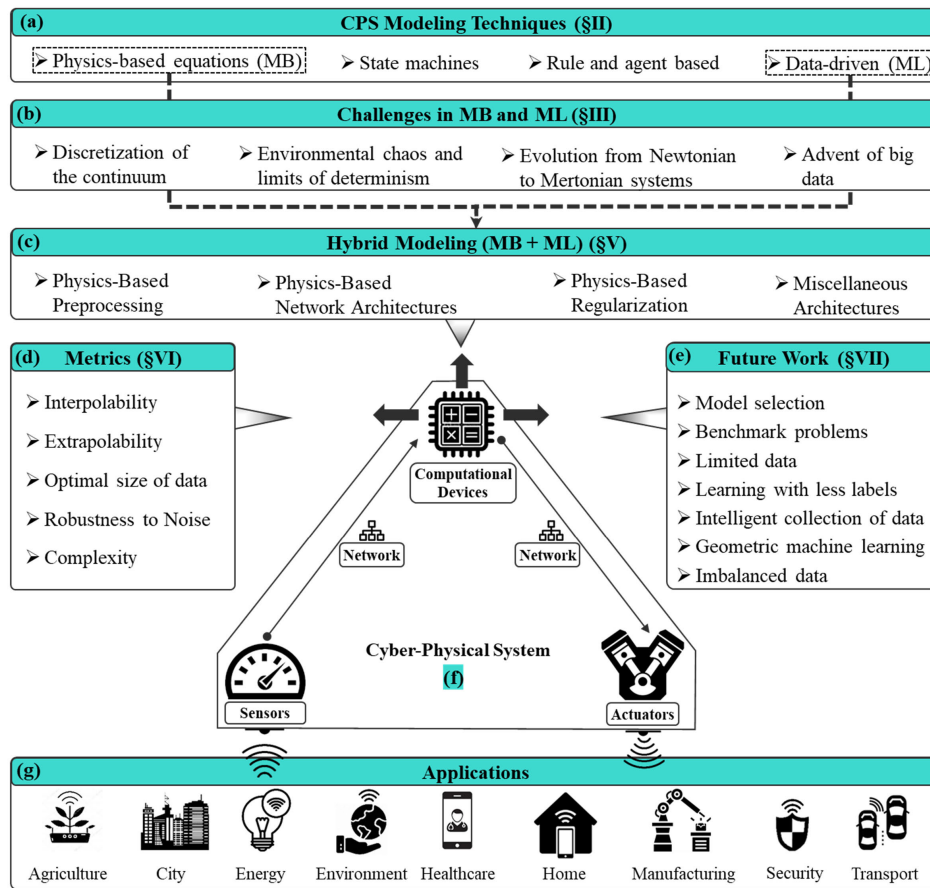


FIGURE 2. Structure of the current outlined work on hybrid modeling in CPS domain: (a) CPS modeling techniques; (b) challenges in the CPS modeling techniques; (c) hybrid models incorporating MB and ML; (d) metrics for evaluating the hybrid models; (e) direction for future work in hybrid modeling; (f) cyber-physical system; (g) applications of CPS.

the segment of the work, which was more manual but less cognitively challenging. Automation powered by artificial intelligence has the potential to capture some part of the remaining segment of the work. It can help augment human capabilities by being capable of doing mental work along with the menial work. While doing so, the CPS has to meet a few additional goals that are still open for research. The CPS must bring the precision, accuracy, and latency from the computing world to the physical world. The asynchrony in time and space must be fixed. The CPS must cater to the privacy and security of itself and the associated users. The intrinsic and extrinsic heterogeneity, complexity must have to be addressed. The internal, external, and cross-functional dynamics must be comprehended and modeled. It must abridge the gap between the discrete and logical functions from the virtual computing world and the continuous physical dynamics of the real world.

Several attempts have been made at modeling CPSs to harness the revolutionizing potential of the CPSs. Depending on the intended goal of the CPS and the characteristics of the CPS, the modeling techniques have broadly diversified into four categories: (1) Physics-based equations; (2) State

machine based; (3) rule and agent based; and (4) data-driven models. All these modeling paradigms are pictorially represented in Figure 3. These modeling techniques encompass white-box models, grey-box models and black-box models [55], [56]. The physics-based equations and state machines are used to model the continuous and discrete dynamics of a system respectively. They are purely based on the physical dynamics of the system. They are robust but lack intelligence. Rule and agent based models leverage the data generated by the system, along with the knowledge of CPS. Hence, rule and agent based models are more capable. The data-driven methods take advantage of data. The following subsections include a brief critical appraisal of these modeling paradigms.

A. PHYSICS EQUATION BASED CPS MODELING

The physical dynamics of all systems, including CPS, are continuous in nature [57]. They evolve with time. Such continuous dynamics are effectively captured by physics-based equations. These physics-based models usually take the form of equations. The equations can be simple dependence equations or ordinary differential equations or partial differential

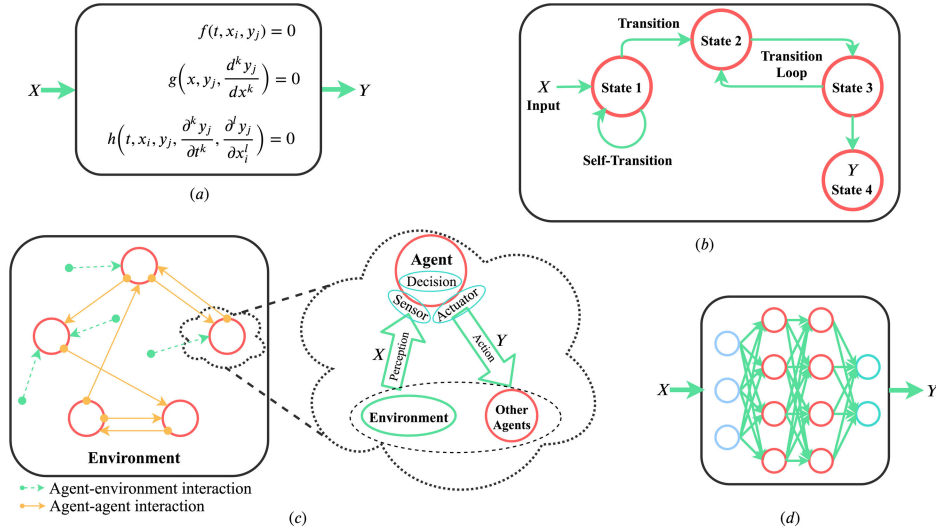


FIGURE 3. CPS modeling paradigms: (a) Physics-based equations; (b) State machines; (c) Rule and agent based models; (d) Data-driven (DNN) models.

equations or a combination of them [58], [59] (Figure 3(a)). For a time(t) dependent system of n independent variables denoted by x_i where $i = 1, 2, \dots, n$ and m target variables denoted by y_j where $j = 1, 2, \dots, m$, a simple dependence equation can be of the form

$$f(t, x_i, y_j) = 0 \quad (1)$$

If the dynamics takes the form of ordinary differential equations [60], then it will be

$$g\left(x, y_j, \frac{d^k y_j}{dx^k}\right) = 0 \quad (2)$$

In case of complex systems, it is difficult to identify the individual impact of a parameter accurately. In such cases, the dynamics takes the form of partial differential equations [61]. Then the equations will be in the following form:

$$h\left(t, x_i, y_j, \frac{\partial^k y_j}{\partial t^k}, \frac{\partial^l y_j}{\partial x_i^l}\right) = 0 \quad (3)$$

where $k, l = 1, 2, \dots, p$ are the indices of the differential operators. The equation-based models are limited to modeling continuous dynamics. Causality, conservation laws, objectivity, and composability are the main attributes of physics-based equations [62].

B. STATE MACHINES BASED CPS MODELING

State machines are used for modeling discrete dynamics [59], [63]–[65]. Discrete dynamics involve a set of discrete sequential steps, e.g., the number of times a ball bounces under gravity. The number is going to be a natural number, which is discrete in nature. A fundamental difference between the continuous and discrete dynamics is that unlike continuous dynamics which is inherently smooth, discrete dynamics is discontinuous. Discrete events are instantaneous in nature. So, the derivative is zero before and after

the event but tends to hit infinity at that instant. Discrete dynamics can be modeled using finite state machines or extended state machines depending on the number of states. They can be event-triggered or time-triggered. Mathematically, a finite state machine is a tuple of five variables (*States, Inputs, Outputs, Update, InitialState*) [9], [65]. State machines can be broadly differentiated into Mealy machines and Moore machines. If the output is produced when a transition happens, it is a Mealy machine [66]. If the output is produced when the machine is in a particular state, it is a Moore machine [67]. They are convertible into each other. A simple hypothetical finite state machine model is depicted in Figure 3(b).

State machines are an efficient way to model discrete dynamics. Causality, time invariance, linearity, stability, and memorylessness are some of the properties common to both physics-based equations and state machine based methods [9].

C. RULE AND AGENT BASED CPS MODELING

CPS can also be represented using a set of agents and semantically defined rules. The state machines are unlikely to capture the interdependencies among different states. Rule and agent based models promise to capture the intra- and inter-dependencies among the cyber and physical components along with the heterogeneity and complexity of the CPS. Such systems are also referred as *multi-agent system* (MAS) [68]–[70] and *agent based models* (ABM) [71], [72]. MAS or ABM describes the system by describing its constituents. As a result, they are quite effective at modeling a whole system when the whole is bigger than the sum of its constituents due to their interactions [71]. MAS is supported by a set of sensors that sense the environment and create data. The data is acted upon by smart *agent(s)* using decision making algorithms (Figure 3(c)). The agents make

decisions based on a semantically defined set of *rules*. As they use data and semantic rules, MAS is capable of extracting insightful information from the data. Temporal continuity, social skills, and autonomy are the main attributes of an agent [73], [74]. These intelligent MAS are not completely black box in nature. They incorporate the recognized semantics of the system into the model. E.g., the semantics addresses the difference between the *flow* in the cyber (data) and physical domains (power in energy systems vs. vehicles crossing a toll booth). A simple example could be a parking lot where the CPS can guide the oncoming vehicles for an empty parking slot and stop accepting additional vehicles if the lot is full.

Unlike the physics-based or state machine based models, which are limited by the need for adaptability and interoperability [74] in CPS, rule and agent-based systems are not limited by them. The former methods are neither smart nor adaptable. But, rule and agent based systems are smart and adaptable. This is why the research work in rule and agent-based models of CPS has seen multifold growth in recent times. Even then, as the rule and agent based models exploit the semantics of the system, complex systems pose a serious challenge to MAS [75] like those in physics-based or state machine based models.

D. DATA-DRIVEN CPS MODELING

The data-driven models completely rely on data. They obviate the knowledge of system dynamics. The data-driven models must be able to model the inherent characteristics of the CPS, i.e., continuous (regression) and discrete (classification) dynamics of the CPS.

Attempts at data-driven modeling of CPS for capturing the hybrid dynamics tried to exploit concepts from bayesian learning [76], algebraic geometry [77], bounded error [78], compressive sensing [79], inference of state transition logics [80], regression approach underpinned by recursive clustering and linear multi-class classification [81], symbolic regression [82], hybrid stable spline algorithms [83] etc. An excellent review of earlier methods of identifying hybrid systems can be found in [84]. The earlier methods [76]–[78], [85] mostly focused on simple piecewise affine systems which use linear transition rules [80]. Recent methods try to capture the non-linear dynamics from relatively complex systems [79]–[83]. Recently, DNNs (Figure 3(d)) with non-linear activation functions received widespread recognition for their ability to learn complex non-linear functions [86]–[89]. They have also started to find use in modeling CPS [90], [91]. Currently, there is a consensus that only models powered by artificial intelligence can achieve acceptable levels of autonomy by coping up with the uncertainties of the real world [92]. The DNNs, which come under the umbrella of artificial intelligence and have taken the canonical form of AI techniques, are best equipped to make the CPS autonomous. Nonetheless, the deep learning techniques are plagued by their own limitations, which are briefly discussed in section IV. The primary reasons for the poor performance of purely ML-based data-driven methods

in CPS modeling for state assessment of the CPS are (1) physical systems work in dynamic environments where system performance is changing continuously, (2) the inability of ML methods to generalize beyond their initial set of training data, (3) sparse training data set, (4) ML methods are agnostic to underlying physics, resulting in predictions that are sometimes inconsistent with the laws of physics, and (5) in predictive CPS maintenance applications, the primary focus is on anomaly detection that can lie in the tail end of the data distribution. Hence, the current work focuses on combining conventional physics knowledge with neural networks (NNs) to overcome those limitations.

III. CHALLENGES IN MODELING CYBER-PHYSICAL SYSTEMS

The models of the CPS must model every component of the CPS. In other words, it should constitute the models of physical components and processes, computational algorithms, software, and networks. Interpretability and composability of the physics-based models are the main drivers behind their widespread adoption. Interpretability prevents failures. Composability facilitates the construction of complex CPS. Together they make the physics-based models of CPS ubiquitous and indispensable. But, the model-based systems are plagued by a diverse set of challenges such as solver dependency of the model, Zeno behavior, non-determinacy, heterogeneity of the system, maintaining the consistency among the components of the model, distributed behavior and misconnected model components [93].

Different numerical solvers used in a CPS model may use different numerical precision, leading to different behaviors of the CPS models [60], [93]. As simple components evolve into complex systems, understanding and modeling the interaction among the components becomes even more challenging. Ensuring consistent evolution among all the components is also difficult. In addition to this, the time lag created during computation and communication must also be modeled. The unevenness in measurements of time across the components, loss in communication, network delays, etc. must also be incorporated [93]. Our goal is not to discuss the methods to address all the above-mentioned challenges. Rather, we mainly focus on specific CPS modeling challenges that can be eradicated or minimized by combining model-based and data-driven methods.

A. DISCRETIZATION OF THE CONTINUUM

In cyber-physical systems, cyber components like microprocessors and digital communication networks portray discrete dynamics. Whereas, the physical elements of the CPS like electro-mechanical, humane, and chemical components portray continuous behaviors [9]. Hence, modeling a CPS can be split into two parts: continuous behaviors and discrete behaviors. The models must address the semantic dichotomy between discrete and continuous behaviors and their interactions. For modeling any CPS, we must integrate continuous behavior with discrete behavior. The computational models,

which are inherently discrete, must model the continuous behavior using some means like numerical approximation or numerical integration [57], [60], [94]. The time precision of the computational models shall allow modeling the discrete dynamics. Hence, the time precision of the computational models should be finer than that of the discrete system dynamics.

However, the time precision of the computational model is fixed and cannot be refined beyond a certain limit. That poses severe challenges to the computational models to model the continuous dynamics portrayed by the physical elements when the continuous events become superfine. When the physical events become superfine, the time period of the events diminish, and uncountably many events happen in a small time interval. For example, consider a ball bouncing freely under gravity. It is subjected to wind drag. And as the collisions are not perfectly elastic in nature, some energy is lost during the collisions. Gradually, the amplitude (bouncing height) of the ball diminishes. After a certain point in time, the amplitudes become so small that it becomes difficult to differentiate two different free falls of the ball. The time period of the falls also reduces considerably. Consequently, the frequency of the events (collisions) shoots infinity. Such behavior is called as *Zeno behavior*, where infinite events occur in a finite time interval [95]. That leads to huge difficulties in modeling a system. The point after which the events can't be modeled any more by the computational model is called *Zeno point*. Although challenging [95], there is some prior work that attempted to determine the Zeno point of a cyber-physical system [96], but there is a dearth of work on modeling Zeno behavior.

B. ENVIRONMENTAL CHAOS AND LIMITS OF DETERMINISM OF MODELS

In addition to the challenge of discretization of the continuum, two more challenges prevent the models of CPS from becoming robust. They are chaotic behavior of the environment and the limits of determinism of the models [95]. Deterministic systems are predictable. The output can be predicted accurately if the inputs, system parameters, and the initial conditions are known precisely. However, most nonlinear feedback systems encounter uncertainties. Even real-time systems that seem robust, non-chaotic, like the standard scheduling algorithms, can exhibit chaotic behavior in the presence of uncertainty [97]. Chaotic behavior is so fundamental that it weakens determinism. Determinism is an invaluable property as it facilitates definitive analysis of the CPS and enables complex and reliable designs [95]. However, deterministic models are never complete. E.g., a single-threaded imperative program is deterministic, if only the output is taken into account. The same program becomes non-deterministic if the time of computation is also taken into account. So, indeterminacy is inherent. Limiting cases will always arise at which the deterministic models are going to fail, and the model will no longer be valid. When the limiting cases are unidentifiable, non-deterministic models

are the only way out. In such cases, Heisenberg's uncertainty principle seems more relevant [95]. However, the scope of Heisenberg's uncertainty principle is limited to quantum mechanics. It does not extend up to Newtonian mechanics. However, for practical cases, even Newtonian mechanics is inadequate. It has to be supplemented by Mertonian laws to make it more practicable.

C. EVOLUTION FROM NEWTONIAN SYSTEMS TO MERTONIAN SYSTEMS

With the advent of data-driven modeling, the proliferation of Merton's systems has drastically increased than those of Newton's systems. It is only because of the close interaction between actual and artificial life. The CPS has become cyber-physical social systems (CPSS) [98]–[100]. Even though some preliminary low fidelity MB exists for the CPS, which can be adapted for modifications, adapting them for processes that lack well-understood first principles (E.g., mobility behavior, advertisement click rates, energy consumption in apartments and systems with human intervention) is infeasible [58]. Modeling such systems has been possible with the support of big data, cloud computing, internet of things, NNs, reinforcement learning, etc. [101]. This evolution can be broadly restated as traditional systems with small data, and big Newtonian laws have taken over by the modern systems with big data but smaller Mertonian laws. The deterministic systems have taken the background. The foreground has dominated by the statistical systems governed by probabilistic laws. Newton's laws exclusively will be inadequate for modeling CPSs. We have to introduce Merton's laws for modeling such things. Merton's systems are governed by Merton's self-fulfilling prophecy. It says a prediction directly causes itself to become true, because of the feedback between belief and action [102], [103]. Such systems are governed by free will. Hence, they cannot be controlled but can be influenced to achieve the desired objective in a probabilistic environment. The uncertainty in the complex spatio-temporal dynamics of the environment creates a chasm between the deterministic model and the actual physical systems. Ordinarily, causality prevails in Newtonian systems. In the case of Merton's systems, causality is a luxury that is almost unachievable with the available resources for complexity, diversity, and uncertainty [101]. The gap can be mitigated by integrating the model-based approaches based on Newtonian laws and the data-driven approaches based on Merton's philosophy.

D. ADVENT OF BIG DATA

It is important to note that physics-based models for modeling a system do not actually need data to model the system. Rather, they need a deeper understanding of the causal relationship of the system's parameters and variables [104]. So, they are effective at modeling any system of simple to moderate complexity. More importantly, they are quite effective at modeling systems with small to medium scale of data. However, recent years have seen a huge surge in the amount of collected data. More data directly translates into

better information about the system. For complex systems, where inferring the causal relationship between the input and output variables could be a daunting task, a large size of data helps. Data harnessing has seen a surge because of pervasive sensing, rapid computational capabilities, the proliferation of the internet and its users. It led to the rise of big data. Big data needs bigger and effective processing technologies. The processing technologies started with model-based algorithms. But, are shifting into data-driven techniques. For a model-based approach, limiting cases will keep on accumulating as more and more data gets collected. More data will cover the state space better. Hence, it will produce accurate and robust models depending upon the effectiveness of the models for processing the data.

CPS is going to be ubiquitous. It is going to touch and alter every part of life and the ecosystem we are living in. They will be pervasive. Data will be created, processed, and acted upon continuously in real-time. Continuous creation of data will shove the CPS into the scope of big data. Big data is characterized by its characteristic V's [105]–[110]. Although significant variation can be observed in the number of V's available in the literature, all of them have the first three V's: *volume*, *variety* and *velocity*. The volume refers to the huge amount of data that needs to be collected, processed, and stored. Variety is the complexity of the data. It can be structured, unstructured, or a mix of both. Typically, structured data can be organized into a tabular form, whereas unstructured data cannot. Velocity refers to the speed at which data is being generated, which in turn needs adequate processing capabilities. Although MB can be effective at handling the velocity, it will be inefficient at handling the volume and variety of big data. The variety increases the dimension of data. Every additional dimension can be ascribed to a new system parameter. That parameter may have an unanticipated influence on the model. That will contribute to increasing the complexity of the MB. Additionally, with a humongous volume of data, the state space will be covered almost completely. Hence, almost all the exceptional cases will be captured by the data. Every exceptional case is potent to uncover a new phenomenon. And, every phenomenon may require a new model altogether. So, a model encompassing all the exceptional cases will become intractable. The solution is data-driven methods. Although they are not likely to be explainable, they will definitely model the state space better.

IV. SYNERGY BETWEEN MACHINE LEARNING AND PHYSICS-BASED MODELS

Both the ML models and physics-based models can be used for system modeling purposes. As stated previously, the MBs do not need data to model the system. They need the understanding of the system dynamics [57]. MBs need data for calibrating their effectiveness. On the other hand, ML models need data for modeling the system. Data without any consideration of the dynamics of the system [58] empowers the ML methods to be applicable to different types of systems. However, data can never be complete because

of the discrete nature of the input parameters and the output variables of the system [111]. Irrespective of whether data acquisition is expensive or not, data can never fill the state space. So, the chasm between the system and the data generated by it is ingrained in ML models. They tend to fail beyond the regime of the training data [58]. Thus lack of extrapolability. This is evident from the success of convolutional neural networks (CNNs) with the development of ImageNet database [112], [113], which were prone to failure with older databases [114], [115]. So, it is wise to conclude that ML models, although accurate, can never be complete. The DNNs operate in a monolithic manner, hence inhibiting modularity, hierarchy, and composability. The whole network learns simultaneously. They only care for learning a mapping from the input to the output variables without any care for the intermediate variables. Additionally, as ML models do not pay any heed to the causal relationship of the system variables, they lack interpretability. As the ML models learn, they are susceptible to learn spurious but accurate mappings from the input data to the output data. That makes ML a double-edged sword.

Contrastingly, MBs have the potential to be complete if all the system dynamics from intrinsic to extrinsic and micro to mega-scale are taken into account. However, such dynamics are insurmountably difficult to capture even with delicate simulations and meticulous observations. They can be expensive in terms of cost, computation, and time. So, some dynamics are often ignored [57]. A few assumptions are made. That simplifies the model of the system by disregarding the impact of those dynamics. That makes the MB incomplete and inaccurate [111]. For example, the impact of wind drag, energy lost due to the non-elastic nature of collisions, the elasticity of the material of the ball, the buoyancy of the medium is often ignored in case of the bouncing ball. Although inaccurate, the MB tend to extrapolate the system behavior [58]. If simulations are carried out delicately and observed meticulously for capturing the dynamics accurately, then the MB can become complex [57]. Complexity compromises tractability. Tractability is of paramount importance in control applications. Even sometimes, the impact of some dynamics is consciously ignored to make the model amenable. Even then, the MB extrapolate far beyond the operating conditions utilized for model validation with reduced accuracy [55].

Often, it is easier to acquire data from a system using simulations than understanding the system dynamics. It is quite easy to fit an ML model to the data originating from the system. Data is sparse. But, ML models discover a continuous function to fit the same sparse data. Hence, they fill the state space. So, although data can never fill the state space, the model can. With the same data, it is easier to validate an MB, provided it is available. But if the data does not include any anomaly, neither the ML can model it, nor the MB can be validated against it. E.g., consider the *magnitude of acceleration* of the ball bouncing under gravity. The magnitude of the acceleration is equal to the acceleration due to gravity when the ball is in air. The ball is in the air during most of its flight.

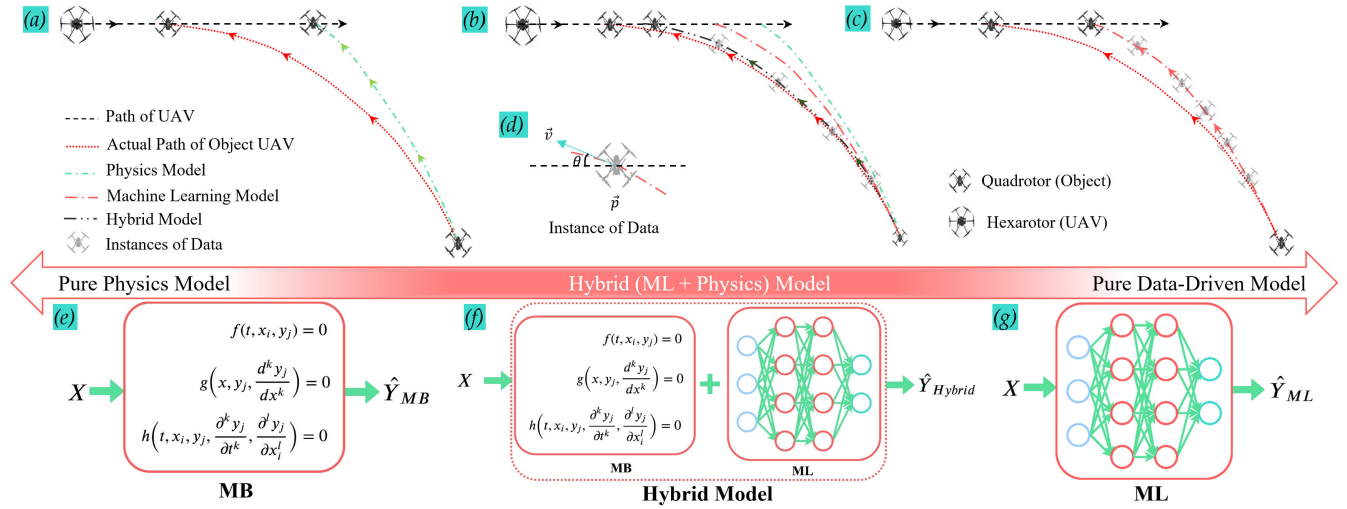


FIGURE 4. The spectrum of physics and data showcasing the effectiveness of pure physics-based, data-driven and hybrid models for predicting the path of an oncoming quadrotor from a hexarotor in an autonomously operating swarm of UAVs for CPS applications. (a), (c) and (b) represent the predicted path of pure physics, pure data-driven and hybrid models respectively. The corresponding models are shown in (e), (g) and (f). Figure (d) shows an instance of data which comprises of the position ($\vec{p}$), orientation ($\vec{\theta}$) and velocity ($\vec{v}$) of the quadrotor.

But, the magnitude is different from the acceleration due to gravity at the moment when the ball hits the floor. It appears as an anomaly in the plot. If the data does not capture the moment, neither the ML nor the MB can guarantee consistent behavior. So, it is essential for the data to capture all the events. The same goes for the bouncing ball when it exhibits Zeno behavior. As the data can't be recorded after the Zeno point due to the limitation of the simulator, ML models cannot model Zeno behavior.

However, the causative agents behind the Zeno behavior can be identified, and MB can be used to model them. The causative agents for the Zeno behavior of the bouncing ball are the inhibiting forces due to the environment. Air drag, the viscosity of the medium, non-elastic collisions, deformable nature of the material of the ball contributes to the inhibiting forces. The initial height of the ball and buoyancy of the medium are the driving forces. These aspects of the system can only be brought into the model using MB. Hence, it is sometimes prudent to complement the ML with MB [58], [111], [116]. By combining them, one can actually recover a complete model of the CPS. We feed the knowledge we have into the MB. The ML model discovers the knowledge which we do not know [117]. Nonetheless, all models are wrong, but some are useful [118], [119]. Hybrid models are comparatively more useful and more efficient as they harness the best of both worlds. The efficacy of hybrid models over pure MB and ML models are pictorially shown in Figure 4 for predicting the path of an oncoming quadrotor from a hexarotor in an autonomously operating swarm of unmanned aerial vehicles (UAVs) used in CPS applications. Additionally, hybrid models can be re-calibrated conveniently. As new data becomes available, the re-calibration of MB can face the difficulties associated with the solving of underlying highly complex optimization problems [120]. Whereas, the

ML gets calibrated in an incremental manner. That eases the re-calibration of a hybrid model.

V. HYBRID MODELING: INFUSING PHYSICS INTO DATA-DRIVEN MODELS

Consider a CPS predictive modeling task such as the path prediction of an oncoming quadrotor from a hexarotor in case of an autonomously navigating swarm of UAVs (Figure 4) for CPS applications. Given input state variables $\mathbf{X}$ (Figure 4(d)), model parameters $\mathbf{P}_{MB}$, and target variable Y (state variables of the quadrotor at a future instant of time), the physics-based models, $f_{MB} : [\mathbf{X}, \mathbf{P}_{MB}] \rightarrow Y$ maps input state variables and parameter values to the target variable Y . Predictive modeling in a purely model-based approach is primarily focused on “calibrating” model parameters $\mathbf{P}_{MB}$ using observational data. Typically, in model-based approaches, we make simplifying assumptions and build simple physics-based models of key system components. For example, we model the response of a mechanical CPS system as a response of a linear mass-spring-damper component, although the behavior is typically nonlinear. Therefore, Y_{Phy} (output of the calibrated model) typically is an incomplete representation of the target variable due to abstraction, simplification, and missing physics in f_{MB} , and this causes the model output Y_{Phy} to be different than observations Y as shown in Figure 4(a). The data-driven machine learning approach, however, is focused on training a model $f_{ML} : \mathbf{X} \rightarrow Y$ over a set of training data to produce estimates of outputs $\hat{Y}$ given inputs $\mathbf{X}$. The ML model also cannot predict the path exactly (Figure 4(c)) as they have limited capability to capture the system dynamics. Both these approaches have their limitations.

In a hybrid modeling approach, one creates a hybrid combination of MB and ML approaches that combine the power of both (Figure 4(b) and 4(f)). One way to combine them

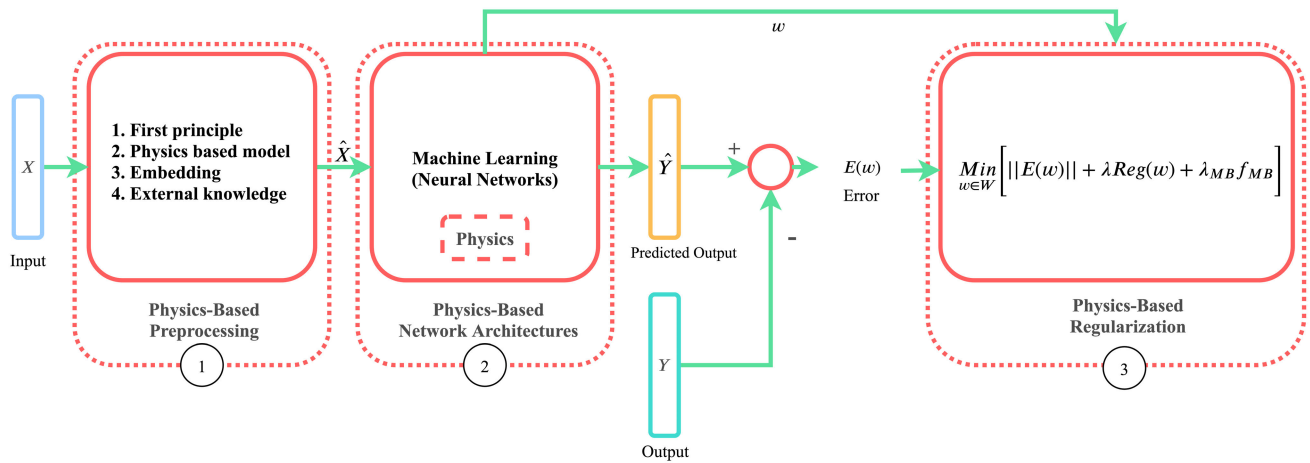


FIGURE 5. Hybrid modeling domain can be categorized into three main classes (1) data preprocessing done by physics-based approaches, (2) fusion enabled by innovative architecture of ML (neural network) model, and (3) development of suitable regularization functions.

is to use the output of f_{MB} as an input to f_{ML} or f_{Hybrid} : $[X, P_{MB}, P_{ML}] \rightarrow Y$, where $f_{Hybrid} = f_{ML} \circ f_{MB}$. The machine learning component will thus encapsulate the remaining unmodeled complexity of the system in a lumped form. In this approach, the ML model acts in a complementary fashion to enable accurate model prediction. If the physics-based models are accurate, then the hybrid model will ensure $Y_{phy} = \hat{Y}$. However, even if the ML model leads to more accurate modeling capability but lacks the mechanistic coherency of the underlying system (for example, violates the physics of the overall system), it cannot be used in the CPS modeling domain. Therefore, we constrain the ML model to learn in a fashion that respects the physics of the underlying process. Additionally, one can use synthetic labeled data from physics-based models to complement the experimental training data and train the ML models to address the data sparsity issue. Furthermore, a tuned f_{Hybrid} output $\hat{Y}$ can be used to generate additional data to fine-tune the parameters P_{MB} (coarsely calibrated initially using sparse observations) of the MB physics equation in an intelligent fashion.

There are two main novel aspects of hybrid modeling approach: (i) the hybrid combination of MB and ML complements each other to create more precise predictive models that respect the physics of the underlying system and (ii) the use of both the physics-based models and sensor acquired data to train the ML model and estimate the parameters of MB model in a symbiotic fashion. There has been several works across a wide variety of applications which attempted at combining the ML and MB models. The terminology for referring such architectures is not consistent. The keywords which were used to refer such architectures include ‘using prior information’ [121], [122], ‘physics guided’ [123]–[127], ‘residual physics’ [128]–[131], ‘physics based’ [132], ‘physics fusion’ [128], ‘physics infused’ [133], ‘theory guided’ [14], [134], ‘hybrid’ [135]–[137], ‘physics informed’ [138]–[143], [143]–[146], ‘physics constrained’ [123], [147], [148] etc. However irrespective of the name of the hybrid model, they

share a lot of commonalities. Depending on the nature of the interaction between the P_{MB} and P_{ML} in a hybrid model, the existing work in hybrid modeling domain can be broadly categorized into three categories: (1) approaches that use P_{MB} as preprocessing or inputs to machine learning model (such as deep learning) (Figure 5①), (2) innovative network architectures that imbibe physics of the problem at hand in P_{ML} (Figure 5②), and (3) loss functions that enforce physics infusion in the learned DL model (Figure 5③). The three major categories are motivated by the control levers of a DNN for learning better functions to obtain accurate mappings. The three levers are better data which can be acquired by better preprocessing techniques, infusing good features into the network, and better loss functions. A single DNN can incorporate all the control levers in an integrated fashion, as shown in Figure 5. We lump other existing work in the hybrid modeling domain that does not fall exactly into one of three categories defined above into a fourth category termed “Miscellaneous”. Additionally, we primarily focus on the work done in hybrid modeling, where the used ML approach is Deep Learning (NNs). However, similar ideas could also be used in conjunction with other ML approaches such as Support Vector Machines (SVM) [149]–[151] and Random Forest (RF) [152], [153] etc. Additionally, as the prior research work on hybrid modeling in the cyber-physical space is quite limited, we open up our scope of the hybrid models to all existing applications to make it comprehensive and complete.

A. PHYSICS BASED PREPROCESSING (PBP)

Almost all the methods for processing data follow three steps: data preprocessing, feature extraction, and feature processing. The sole purpose of preprocessing is to ease the use of data in the succeeding steps. Noise removal, data transformation, data reduction, instance selection, dimension reduction, etc. constitute the preprocessing techniques [154]. Unlike the feature extraction techniques, which could be very specific,

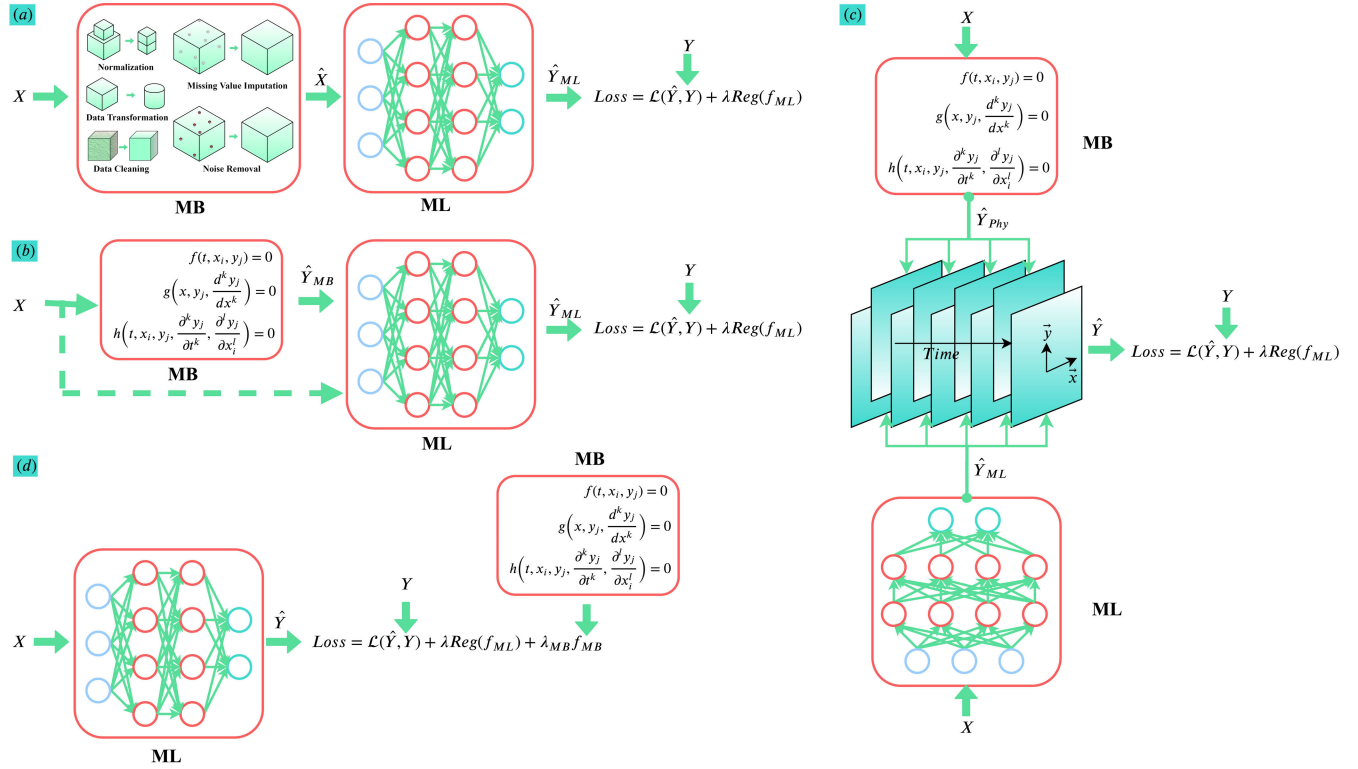


FIGURE 6. Different types of hybrid physics-guided machine learning models: (a) Physics-based preprocessing; (b) Physics-based network architectures that ingest the output of physics models with or without the raw data; (c) Physics-based network architectures for spatio-temporal data where the $\hat{Y}_{MB}$ and $\hat{Y}_{ML}$ are deeply integrated into each other; (d) Physics-based regularization.

the preprocessing techniques can be used in a wide variety of applications [155]. Preprocessing is essential for ML models. Normalization is the simplest form of preprocessing. Band-pass filtering, downsampling [156], channel selection, time window setting [157], data cleaning, encoding [158] are a few more examples of simple preprocessing techniques. In the first class of hybrid architectures, the MB is used for preprocessing the data before feeding them into the ML model. The MB preprocesses the input data x and converts it into $\hat{x}$. The ML learns a mapping between the preprocessed data $\hat{x}$ and the actual outputs y and predicts the actual output as $\hat{y}$ with an error ϵ_i given by $\epsilon_i = y - \hat{y}$. Such physics-based preprocessing hybrid models are illustrated in Figure 6(a).

As the DNNs are capable of learning from pristine data, they have almost eliminated the need for data preprocessing. However, the elimination of data-preprocessing comes at the cost of deeper and complex networks. Again, insufficient and noisy data adds to the pain of the complexity of the ML models. Hence, preprocessing can help to alleviate the pain. Raw data is usually a time-series signal. Techniques like fast Fourier transform (FFT) can be applied to convert it into the frequency domain. Consequently, the features also get transformed. Ordinarily, frequency domain representations are used for stationary signals. Due to their high sensitivity, they are widely used for anomaly detection tasks [159]. For non-stationary signals, time-frequency representations are used.

The time domain and frequency domain representations are one-dimensional representations. Wavelet transform, empirical mode decomposition, short-time Fourier transform can be used to convert the 1D representations into 2D time-frequency domain representations [160].

The preprocessing techniques for data analysis can be broadly split into time-domain analysis, frequency domain analysis, and time-frequency analysis. Mean-variance, empirical mode decomposition [161], and kurtosis estimation fall under time-domain analysis. Fast Fourier transform, bispectrum analysis are examples of frequency domain analysis. Short-time Fourier transform (STFT), wavelet transform [162], [163], Hilbert-Huang transform [163], sparse decomposition [164], wavelet packet transform comes under the umbrella of time-frequency based preprocessing techniques [165].

Hybrid architectures which use physics-based preprocessing have found applications in manufacturing industry [166]–[169], healthcare industry [170]–[172], energy industry [173], and for processing hydrological data [174]–[176], meteorological data [177] among several others. Compressed sensing was used for extracting low dimensional features from raw temporal data before passing it into a stacked denoising autoencoder DNN [178]. Other preprocessing techniques like spectral kurtosis [179] and envelope analysis [180] were used for extracting the frequency band

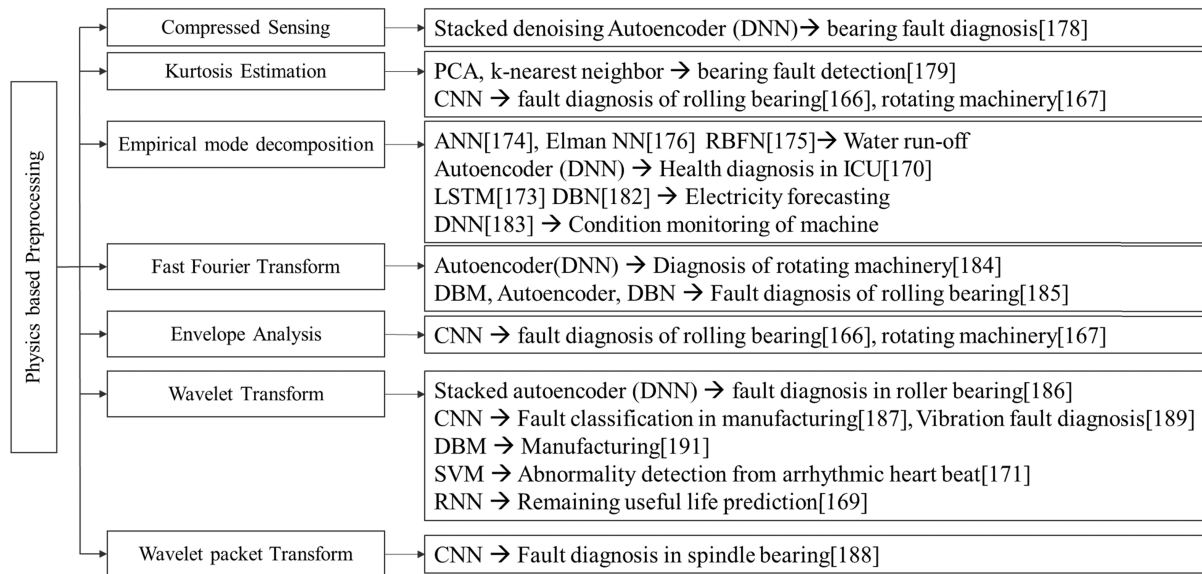


FIGURE 7. Different cases of physics-based preprocessing models .

with the most useful information. The output was further processed by using fast Fourier transform before feeding it to a CNN [166], [167]. Ensemble empirical mode decomposition (EEMD) [181] was used for decomposing the input data into a set of intrinsic mode functions (IMF) and a residual series [174]. Each IMF was then supplied to an ANN for further processing. Data preprocessed with empirical mode decomposition (EMD) were supplied to a radial basis function network [175] and Elman NN [176], to an autoencoder [170], into a long short-term memory (LSTM) network [173] and a deep belief network (DBN) [182]. EMD was also used for extracting low-level features before proceeding to a DNN for extracting high-level features [183]. The frequency spectrum of raw data was fed to a DNN based stacked auto-encoder [184]. Time-frequency domain data and energy spectrum features of the data were used for preprocessing before feeding the data into deep Boltzmann machines, stacked auto-encoders, and deep belief networks [185]. Non-linear soft threshold and digital wavelet frames were used to denoise the data before feeding it into a stacked auto-encoder [186]. Wavelet analysis was used for converting raw data into 2D time-frequency images, which were subsequently fed into a deep CNN [187]. Data transformed as wavelet packet energy image was supplied to a CNN [188]. The signal was decomposed using Morlet wavelet to get a multiscale spectrogram image [189]. The spectrogram image was passed into a CNN for further processing. Deep Boltzmann machines usually accept data preprocessed by wavelet transform or Teager-kaiser energy operator [190], [191]. A computationally inexpensive delayed error normalized least mean square (DENLMS) [171] filter was used for preprocessing the data [171]. Additionally, discrete wavelet transform was used for extracting higher-level features like heart rate variability (HRT) before passing the data to the ML

model [171]. The short-time Fourier transform is computationally less expensive than wavelet transform and Hilbert–Huang transform. Discrete wavelet transform was used for removing noise before feeding the data to a recurrent neural network (RNN) [169]. Figure 7 summarizes this class of hybrid models. The impact of a few preprocessing techniques like PCA, scaling, and normalization on the performance of autoencoders were studied in [192]. Several preprocessing techniques like envelope metrics, statistical metrics, frequency, and discrete domain metrics were reviewed in detail by [193] for feature extraction. For a comprehensive review of preprocessing techniques [194] can be referred.

Although the preprocessing techniques encapsulate a wide variety of goals like noise removal, data transformation, feature extraction, most of them are shallow features. They do not add any system-specific information to the ML model.

B. PHYSICS BASED NETWORK ARCHITECTURES (PBNA)

The PBNA tries to explore the best combination/fusion of NN architecture and data emanating from the MB component. A plethora of such architectures has surfaced recently. The non-existence of the answers to the questions ‘how do the NN actually learn the features of a system?’ and ‘what is the best architecture for introducing MB into an ML model?’ are providing the thrust for such architectures. The quest has germinated into a new research domain called neural architecture search [195]–[198]. Broadly, the architectures can be one or a combination of DNNs, CNNs, RNNs, and other NNs with physics. The innovative architectures can ingest the output of the MB into a particular node of a specific layer of the ML (Figure 6(b)) or the output of the MB can be embedded into the NN (Figure 6(c)). In addition to the output of the MB, they can also take raw input data as another input (Figure 6(b)).

Such physics-based architectures have found use in environmental data analysis [123], dynamical systems [133], biological systems [133] and image processing [132]. The output of the physics model was used as an additional input feature for a multilayered perceptron for modeling the temperature of a lake [123]. The output of the MB was fed into the last layer of an LSTM network along with the unprocessed inputs for modeling an inverted pendulum and growth of cancer cells [133]. The physics-based motion model of aerial vehicles was sandwiched between two RNN modules for multistep prediction of the non-linear dynamics of the aerial vehicles [137]. It enhanced the training and convergence speeds in addition to the accuracy of predictions. A data augmented residual model, and a recurrent data augmented residual model [131] were formulated for creating a physical simulator offering universal uncertainty estimates. In TossingBot [129], the output of the physics-based controller was fed to FCN ResNet-7 [199] for robotic applications. A deep lagrangian network (DeLAN) [200] was developed for learning the motion of mechanical systems like robot tracking control. It incorporated Lagrangian mechanics into the hybrid architecture. The fusion methods were categorized into series and parallel models depending on the way the MB and ML are connected [120], [201], [202]. A combination of series and parallel connections was used for predicting the failure of a turbine blade [201]. The MB was fused with an ANN and SVR [203] for modeling rainfall-runoff. The work in [132] attempted to recover the shape of an object by letting a CNN based autoencoder to figure out a way for combining the physical solution and polarized images of the object. A tensor basis neural network [204] was proposed where rotational invariance was fed to the last hidden layer of an MLP in the form of a tensor for modeling Reynolds stress anisotropy. An adaptive neuro-fuzzy inference system [205] was infused with the predicted tool wear for prediction of remaining useful life [136].

The architectures discussed so far directly fuses the output of the MB into the ML model. But, such direct fusion can't be done in case of spatio-temporal data and accompanying spatio-temporal physics models. Such data are encountered in videos, fluid, and thermal applications. In such cases, the output of the MB is deeply integrated into the ML model by embedding the output of the MB model at all steps of time and space (Figure 6(c)). Such deep embedded hybrid models have been used in fluid flows [139], [140]. The outputs of MB was fed into a NN as a set of collocation points. The number of collocation points can be increased to increase the influence of physics in the hybrid architecture. They developed the architecture for both continuous and discrete-time models. This architecture was used for both solving [139] and discovering [140] nonlinear PDEs. Invariance or symmetry attributes were embedded into a NN [206] by determining the invariant basis and training the NN on them. It was used for turbulence modeling and solid mechanics.

In addition to these, various works tried to encrypt the MB as a NN and fuse it with an ML model. An MB

transformed as a NN coalesces better with a NN. A cellular NN [207] was used to solve the PDE in Hybrid-Net [135] before attaching it to a ConvLSTM network for fluid and thermal applications. A symbolic multilayered NN inspired from equation learner [208], [209] was used for blending prior knowledge in [210] for learning accurate PDEs from data. In [211], the *LEarned Alternating Direction Method of Multipliers* (Le-ADMM) [212] method is unrolled into a NN for solving inverse problems. It was appended to another pure NN or U-Net [213] for reconstructing mask-based lenseless images into traditional images from lensed cameras. Such a physics-based unrolled network has also been used for image reconstruction [214] and optimizing coded-illumination [215] in case of a LED array microscope.

As figuring out the best architecture for a particular problem is still an unsolved problem, a physics-based neural architecture search [128] model has been developed for predicting the trajectory of a projectile and for estimating the velocities of rigid objects after a collision.

C. PHYSICS BASED REGULARIZATION

In this class of hybrid models, the physics-based equations are used as an additional regularization term in the loss function of the NN. The NN is constrained to acquiesce to the laws of physics. The principle behind this class of hybrid architectures is penalizing/minimizing the violation of physics-based constraints or governing equations. The physics equations $y = f(x)$ are restated as $y - f(x) = 0$ for using them as a constrained loss. Minimizing the loss $y - f(x)$ becomes an additional goal in the loss function (Figure 6(d)). The physics-based pre-processing and fusion architectures need the MB to be evaluated or require the PDEs to be solved. But, regularization based approaches directly engulf the equation as a constraint, thereby altering the computational cost.

Such hybrid models have been used for analysis of environmental data [123], machine vision applications [148] and fluid dynamics [216]. Temperature-density and density-depth relations were used as constraints for modeling the temperature of a lake at different depths [123]. A weighted constraint function was used to penalize data that are inconsistent with prior knowledge in case of the video of a projectile [148]. The loss function of the *pix2pix* architecture in a conditional GAN [217], [218] was penalized by using a constraint enforcing module for spatial sensitivity analysis for predicting urban land use [219]. The loss of a fully connected NN was made to include boundary conditions and residuals of the governing equations for surrogate modeling in hemodynamics [220]. Reinforcement learning was also used for learning non-parametric models under constrained state spaces in continuous environments [221]. It was used for dynamic applications like quadcopter navigation and cart-pole problem. Physics has also been used as a prior in reinforcement learning [222]. As low fidelity MB's capture the general trend and the high fidelity MBs additionally capture intricate local details, a multi-fidelity physics constrained NN was proposed for incorporating linear and nonlinear PDEs [223].

Approximation and monotonicity constraints were also incorporated into the loss function of a NN [224] for predicting oxygen solubility in water.

Various regularization based hybrid approaches were used in case of availability of no or less labeled data. A constrained convolutional autoencoder was used for surrogate modeling and uncertainty quantification in steady-state flow applications [147]. An architecture with both fusion and constrained regularization was used in [225] for multiphase flows. It also included an additional loss term for the difference between the output of MB and the actual output. It performed well under data scarcity. A NN was trained with a video dataset without any labels in a supervised manner purely by using a physics-based constraint [148].

D. MISCELLANEOUS ARCHITECTURES

Accurate inference of the physical dynamics of a system by using conventional principles may need mountainous efforts even with plenty of data. The ML models only learn a mapping between the values of the input and output variables. So, neither the NN based ML models nor the hybrid architectures discussed so far can learn new interpretable physics. It is because of two factors. First, the data by itself is meaningless if not underpinned by physical dynamics. E.g. compare the ball bouncing on earth ($g_{\text{earth}} = 9.8 \text{ ms}^{-2}$) with the same ball bouncing in a viscous medium on moon ($g_{\text{moon}} = 1.625 \text{ ms}^{-2}$). The energy lost while bouncing on earth is purely due to the gravity of earth. On the moon, it will split into the energy lost due to the moon's gravity and the viscosity of the medium. Consider that the viscosity of the medium is controlled on the moon for imitating the ball's bouncing behavior on earth. Now, if the data is fed to ML models to model the dynamics, the ML models can learn the same mapping in both cases. It is because the ML models disregard the underlying physics. In the moon's case, the ML model can ignore the effect of viscosity, which accounted for the difference between the accelerations due to the gravity of the moon and the earth. So, the data doesn't know what it represents. Secondly, the activation functions used by NNs have abysmal physical significance. Hence, the unknown physics which they are capable of discovering is perceived with huge skepticism. So, several data-driven *physically meaningful* ML methods were developed for discovering physics from data. They use physically significant mathematical operations or interpretable activation functions or use classical NNs to unravel the physics-based equations underlying the system dynamics. In such cases, data can be used to extract physics, and the same physics can be used for complementing the ML models.

Sparse Identification of Nonlinear Dynamics (SINDy) [235] used a thresholded least square-based approach to establish a relationship between the target function and a library of candidate functions consisting of algebraic and transcendental functions. Dynamic Mode Decomposition (DMD) [236], [237] built on proper orthogonal decomposition (POD) can be used to extract

spatio-temporal correlations. The DMD tries to infer a linear operator at least for small time periods that approximates the spatial data, which can have millions of degrees of freedom. Koopman operator [238], an infinite-dimensional linear operator, was used for extending DMD for nonlinear dynamics [237], [239]. The inefficiency of linear operators for modeling the non-linear dynamics led to extended dynamic mode decomposition (eDMD) [240], [241]. SINDy, DMD, and Koopman theory has also been extended for control applications in [242]–[244]. The conservation laws of a dynamic system were discovered by symbolic regression [245].

In addition to the above, the equation learner [208], [209] used interpretable activation functions (sine, cosine, identity) in a DNN for learning the governing equations of the system dynamics. PDE-Nets [210], [246] exploited the correlation between convolutional filters and differential operators for learning the PDEs governing thermal dynamics. An n -gram model and a recursive neural network were used in an attribute grammar framework for discovering efficient mathematical identities [247]. Two DNNs were used for discovering non-linear dynamics in terms of non-linear PDEs for spatio-temporal data [248]. These techniques have been used to model dynamic and fluid systems. Variational Autoencoders are known to encode features like physical properties. The discriminator of a GAN can learn a feature level similarity metric. So, in 3D-PhysNet [249], a deep variational encoder with a discriminator was trained using the data from a finite element simulator for discovering the behavior of deformable objects under external forces.

In other systems where the dynamics of some components are unknown, a NN can be used to learn the dynamics of those components. The NN can be used as a module along with the modules corresponding to other components. In Neural Lander [250], the dynamics of a UAV was learned using NN, and the MB was used for its control. A physics infused DNN was used for learning each feature and then concatenated them for predicting the properties of organic compounds [251].

VI. METRICS FOR HYBRID MODELS

Error metrics are essential components for evaluating the efficacy of any numerical modeling scheme. The intention of this section is to provide a summary of a variety of performance metrics that are suitable for the CPS hybrid modeling domain. The outlined summary will help improve general understanding of metrics for hybrid modeling and facilitate their selection in different applications, including the CPS domain.

A. UNIVERSAL ERROR MEASURES

The current section discusses the errors that are fundamental to the holistic evaluation of a model. Consider a collection of data $S = (X, Y)$ where each sample is represented by (x_i, y_i) where x_i represents the input variables and y_i represents the output variables. X and Y represent the set of input and output

TABLE 1. A set of error measures.

Name	Definition	Explanation
Mean absolute error (MAE) [226]	$\frac{\sum_{i=1}^n \hat{y}_i - y_i }{n}$	$\hat{y}$:predicted values; y : true value; n :sample size
Root mean square error (RMSE) [226]	$\sqrt{\frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{n}}$	$\hat{y}$:predicted values; y : true value; n :sample size
Mean squared error (MSE) [227]	$\frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{n}$	$\hat{y}$:predicted values; y : true value; n :sample size
Maximum error [227]	$\max_{i=1}^n \hat{y}_i - y_i $	$\hat{y}$:predicted values; y : true value
Mean absolute percentage error (MAPE) [227]	$100 \times \frac{\sum_{i=1}^n \frac{\hat{y}_i - y_i}{y} }{n}$	$\hat{y}$:predicted values; y : true value; n :sample size
Root Mean Squared Logarithmic Error (RMSLE)	$\sqrt{\frac{\sum_{i=1}^n (\log(y_i + 1) - \log(\hat{y}_i + 1))^2}{n}}$	$\hat{y}$:predicted values; y : true value; n :sample size
Median absolute error (MedAE) [228]	$\frac{\sum_{i=1}^n \text{median}(\hat{y}_i - y_i)}{n}$	$\hat{y}$:predicted values; y : true value; n :sample size
Pearson correlation coefficient [229]	$\frac{E[(y - \mu_y) * (\hat{y} - \mu_{\hat{y}})]}{\sigma_y * \sigma_{\hat{y}}}$	E : expectation; $\mu_{\hat{y}}$: mean of $\hat{y}$; $\sigma_{\hat{y}}$: standard deviation of $\hat{y}$
Adjusted R^2 correlation coefficient [227]	$1 - \frac{n-1}{n-p} \left(\frac{SSE}{SST} \right)$	p :regression coefficients number
Sum of squared regression (SSR)	$\sum (\hat{y}_i - \bar{y})^2$	$\hat{y}_i$:predicted value; $\bar{y}$: mean value of $\hat{y}$
Sum of squared total (SST)	$\sum (y_i - \bar{y})^2$	y :true value; $\bar{y}$: mean value of y
Sum of squared error (SSE)	$\sum (y_i - \hat{y}_i)^2$	$\hat{y}$:predicted values; y : true value
R^2 correlation coefficient [227]	$\frac{SSR}{SST}$ or $1 - \frac{SSE}{SST}$	$0 \leq R^2 \leq 1$; n :number of observations
Deviation evaluation	$\frac{y - \hat{y}}{\hat{y}}$	$\hat{y}$:predicted values; y : true value
Statistical moment-based metric (SMM) [230]	$\left(\frac{1}{\binom{n}{m}} \left[1 - \sum_{1 \leq i \leq m} \left 1 - \frac{\hat{E}[y_i]}{\hat{E}[\hat{y}_i]} \right ^p \right] \right)^{1/p}$	$\hat{E}$: expectation of samples; m :margin interaction
U-pooling(area metric) [230]	$\int_{-\infty}^{+\infty} F_{y_i}(y) - F_{\hat{y}_i}(y) dy$	F :cumulative distribution; $\hat{y}$:predicted values; y : true value
Frequentist's Metric [231]	$\hat{e} - t_{\alpha/2}(n-1) \left(\frac{s}{\sqrt{n}} \right),$ $\hat{e} + t_{\alpha/2}(n-1) \left(\frac{s}{\sqrt{n}} \right)$	$\hat{e}$:estimated prediction error; n : physical observations s :estimated standard deviation;
Bayes factor [232]	$\frac{P(m_1 x)/P(m_2 x)}{P(m_1)/P(m_2)}$	m_1, m_2 : two models; x : data; $P(m_i)$ is prior; $P(m_i x)$:marginal distribution
Akaike information criteria (AIC) [233], [234]	$2k - 2\ln(\mathcal{L}(x, \hat{\mu}))$	$\mathcal{L}(x, \hat{\mu})$:likelihood function; k : parameter number; $\hat{\mu}$:best-fit parameter values for the data
Bayes information criteria (BIC) [234]	$-2\mathcal{L} + k\ln(n)$	$\mathcal{L}$:likelihood function; x : observed data; k : parameter number n :data points number

variables respectively. Both x_i and y_i can be multivariate. The model learns a mapping $f_{Hybrid} : x_i \mapsto y_i$ defined as $f_{Hybrid}(x_i) = \hat{y}_i$ such that $\hat{Y} \approx Y$ with minimum total error E . The total error $E (= \sum_{i=1}^n \epsilon_i)$ is the sum of the individual errors ϵ_i , given by $\epsilon_i = y_i - f_{Hybrid}(x_i)$. There are several ways to define the total error. A detailed list summarizing several measures of error are outlined in Table 1. The mean square error (MSE) and the mean absolute error (MAE) are the most prominent and commonly used measures of error among all of them. The MSE and MAE are defined below:

1) MEAN SQUARE ERROR

The mean square error has taken the canonical form among all the loss functions. Mean square error is the mean of the sum of squares of the difference between the actual and the predicted values. It is defined as

$$MSE = \frac{1}{n} \sum_{i=1}^n [y_i - f_{Hybrid}(x_i)]^2 \quad (4)$$

It is closely related to the quadratic loss function. It is an even, continuous, and differentiable function. The MSE differentiates the error due to bias and variance in a smooth manner. It is the sum of the variance and the square of the error due to bias. Addressing the error due to bias and variance has been a prime cause in statistical data analysis.

The cross-validation technique widely used in machine learning is based on minimizing the overall error by reducing individual errors due to bias and variance. This is why the MSE has taken the canonical form.

It heavily penalizes the outliers. That is good in the case where failure can be fatal or expensive, e.g., manufacturing, autonomous driving, security, etc. However, if only MSE is used as a metric to accept or reject a dataset, a single distant outlier can outright reject an almost perfect dataset.

2) MEAN ABSOLUTE ERROR

The mean absolute error is the main contender against the mean square error for taking the canonical form. It is the mean of the sum of the absolute values of the difference between the actual and predicted values. It is defined as

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - f_{Hybrid}(x_i)| \quad (5)$$

It penalizes all the errors evenly. The MAE doesn't split into errors due to bias and variance directly. It is not differentiable everywhere in its domain. That limits its applicability.

Due to the evident advantages of MSE over MAE and other error measures, MSE has been used to illustrate the ideas in the subsequent sections.

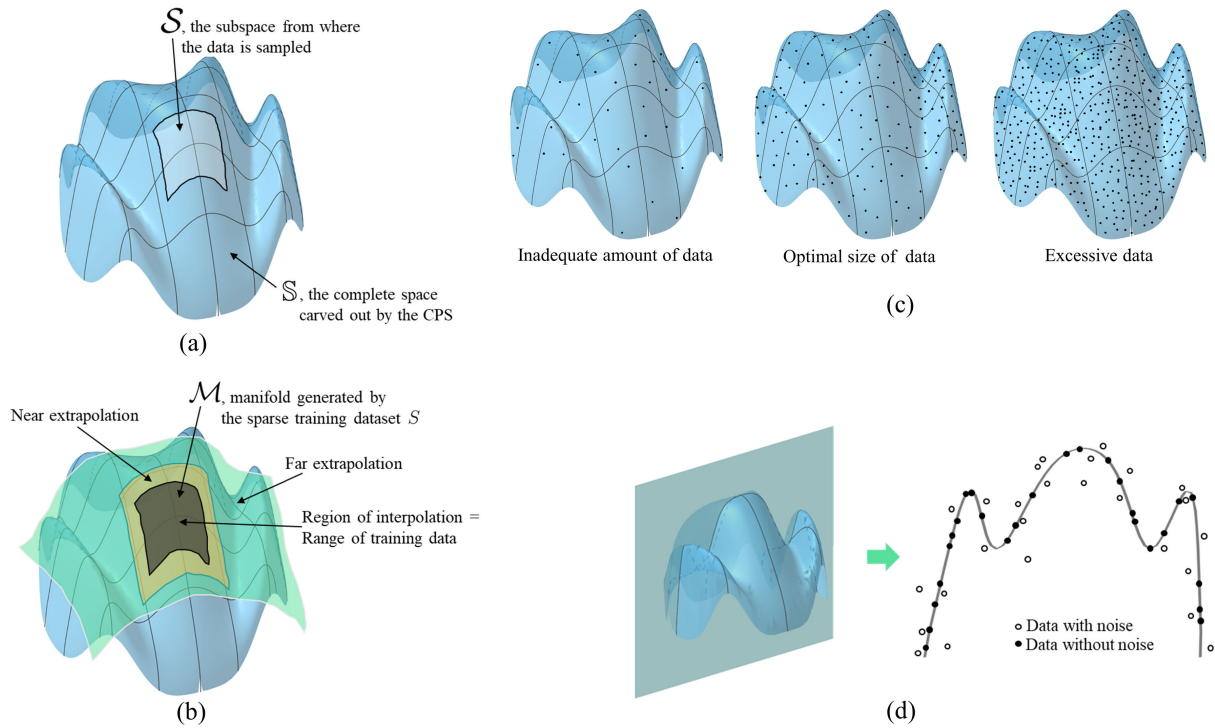


FIGURE 8. The relation between different metrics and the system's space: (a) the subspace from where the data is sampled from the complete space generated by the CPS; (b) Near and far extrapolation from the manifold generated by the hybrid model; (c) Optimal size of data required for modeling the CPS; (d) noisy data.

B. UNIVERSAL METRICS OF A MODEL

Is training a hybrid model on the complete dataset really necessary when the speed of generating data has already surpassed the capacity of storing data [252]? If not, how much data the model actually needs for optimal performance? As noise is inseparable from the data, how to evaluate the robustness of the model to noise? Accuracy is an indispensable metric for every model. CPS is no exception. But for a CPS, the latency between the stimuli and response is also of significant importance. So, how to gauge the latency? The answers to all these questions lie in a systematic set of evaluation metrics.

If we disregard the system-specific evaluation metrics, up until now, all the models were only evaluated for accuracy within the range of the training dataset. Holistic evaluation has never been a priority. Even the cross-validation technique used for model selection in the case of NNs evaluates the model within the range of the training data. As the model can never be trained on the complete space generated by the CPS, can the same model perform well beyond the range of the training data? We propose five metrics for the evaluation of a hybrid model used for modeling CPS. They are as follows:

1) INTERPOLABILITY

The model learns a continuous function f_{Hybrid} from the sparse dataset $\mathcal{S}$. So, the complete manifold $\mathcal{M}$ derived from the sparsely populated dataset $\mathcal{S}$ may not exactly represent the subspace $\mathcal{S}$ of the space $\mathbb{S}$ that is actually carved

out by the system (Figure 8(a) and 8(b)). Mathematically, $\mathcal{S} \mapsto \mathcal{M} \approx \mathcal{S} \subset \mathbb{S}$. Hence, the *interpolability* metric is used to evaluate how accurately the manifold $\mathcal{M}$ represents the system's subspace $\mathcal{S}$. It is defined as the mean of the sum of the squares of the difference between the predicted outputs and the actual outputs *within the range of the training data* (Figure 8(b)). Mathematically, the interpolability metric can be defined as

$$MSE_{Int} = \frac{1}{n} \sum_{i=1}^{i=n} [y_i - f_{Hybrid}(x_i)]^2, x_i \in Range(\mathcal{S}) \quad (6)$$

A lower error translates into better accuracy of the model against unseen data within the range of the training set. Most deep learning techniques conclude their evaluation with the interpolability metric. That's why the earlier deep learning architectures failed drastically against unseen datasets until the development of ImageNet database [112], [113], which included data from diverse categories. Still, they are vulnerable to failure against unseen data. Their success can be attributed to the diverse data they have been trained upon, which ultimately reduced the amount of data they can ever be exposed beyond the space of the training data.

2) EXTRAPOLABILITY

The data is always sparse and is likely to extend its horizon as more and more data is collected. As all the data can never be collected, it is wise to evaluate every model if it performs well beyond the range of the training data $\mathcal{S}$. It is defined as

the mean of the sum of the squares of the difference between the predicted and the actual outputs *beyond the range of the training data* (Figure 8(b)). Mathematically, the extrapolability can be defined as

$$MSE_{Ext} = \frac{1}{n} \sum_{i=1}^{i=n} [y_i - f_{Hybrid}(x_i)]^2, x_i \notin Range(S) \quad (7)$$

The interpolability metric and the extrapolability metric are mutually exclusive. A model with good interpolability and extrapolability can be said to be complete as it can perform well within and beyond the boundaries of the dataset. The interpolability and the extrapolability metrics together constitute the generalizability metric. However, the extrapolability of a model of a CPS is quite difficult to evaluate as it will be difficult to identify the exceptional behavior of the system, simulate it, and collect the extraneous data. A simple example could be expecting a model that has been trained for classifying the predecessors of human beings (e.g., monkeys, apes, and gorillas) to perform well on classifying human beings. The hybrid models have portrayed excellent extrapolability [133], [200], [215], [224], [225], [253].

3) OPTIMAL SIZE OF DATA

Every dataset has different local and global behaviors. When data is sparse, it captures the global behavior. When data is abundant, the local behavior is also captured along with the global behavior. The local variation can be due to noise in the system. So, continuing collecting data to capture the local behavior when you are actually capturing noise is unwise. But, as no clear demarcation exists between local and global behavior, it is difficult to know when to stop harnessing data.

More data doesn't guarantee better information about the system. More data doesn't even improve the model when the accuracy of the model has already stagnated. The performance of a model is sensitive to the size of data until it saturates. It saturates when the complete global behavior is captured. As harnessing data can be expensive, continuing amassing data even after the stagnation of the accuracy of the model is imprudent. So, the accuracy of the model MSE_{Int} shall be evaluated on a running basis, and the data harnessing from the same subspace shall be stopped once the accuracy plateaus. Once the MSE_{Int} converges, new data shall be sampled beyond the convex hull of the training data. It can be mathematically described as

$$MSE_{Boost} = \frac{1}{w} \left[\sum_{i=p+w}^{i=p+2w-1} [y_i - f_{Hybrid}(x_i)]^2 - \sum_{i=p}^{i=p+w-1} [y_i - f_{Hybrid}(x_i)]^2 \right] \leq \varepsilon \quad (8)$$

where w is the window size, p is a random point, and ε is the convergence parameter. If the boost (improvement) in MSE is insignificant ($MSE_{Boost} \leq \varepsilon$), then sampling

data from the current subspace shall be stopped and moved to a new subspace beyond the convex hull of the current subspace (Figure 8(b)). The running average is equivalent to 1-dimensional convolution with average pooling. Hence, the MSE_{Boost} is equivalent to the difference in the pooled averages of 1-dimensional convolutions. The size of the data at the stopping point is the optimal size of data required for modeling the system in the current subspace (Figure 8(c)).

4) ROBUSTNESS TO NOISY DATA

As hybrid models equip ML models, they are not insensitive to noise in the data. So, the model must be evaluated against noisy inputs. Noisy inputs tend to affect the local behavior of the model. The chaos of the environment could be a factor behind noisy data. Artificial noise (Figure 8(d)) shall be introduced into the system during simulations to create noisy data. And the performance of the model must be evaluated against those noisy data. For an artificial noise $\mathcal{N} \sim (\mu, \sigma^2)$ where μ and σ^2 are the mean and variance of the noise respectively, the MSE is defined as:

$$MSE_{\mathcal{N}} = \frac{1}{n} \sum_{i=1}^{i=n} [y_i - f_{Hybrid}(x_i + \mathcal{N})]^2 \quad (9)$$

The noise may or may not have a zero mean.

The deployed CPS is likely to encounter noisy data. So, the CPS models must be robust against the impact of noisy data. Robust models are those that are resistant to noise. Hybrid models have been proven to be more robust to noise than pure MB and ML methods [133].

5) MODEL COMPLEXITY

The performance metrics discussed so far are measures of the performance of the model. They are sufficient for modeling offline systems. But, for online, deployable systems like CPS, additional performance metrics must also be included. An accurate CPS with a delayed response is not desirable. So, additional metrics for gauging the delay, which is mostly due to the complexity of the model, must also be included. Model complexity translates into the time of processing and memory requirements. A complex computational model will require higher memory and time for processing. The computation time can be estimated based on the number of floating-point operations (FLOPs). As CPSs are live systems, complex computational models can delay the response time of the CPS with the environment. Memory is required for storing the parameters of the model, e.g., weight matrices, bias matrices, and physics parameters. Bigger, deeper networks and complex physics models tend to need more parameters and thus necessitate more memory. Equipping miniature mobile CPS with huge memory may not be feasible. So, If memory requirement is high, deploying them online can be difficult. FLOPs and memory can serve as good implementation metrics.

VII. DIRECTIONS FOR FUTURE WORK IN HYBRID MODELING DOMAIN

In CPS, multiple simple components combine for a common bigger cause. Their combination compounds their interaction and impact. Consequently, their individual limitations diminish, and new challenges arise. The new challenges could be due to their interaction or because of any individual component which subverts others. Similarly, in a hybrid model, the ML subverts several shortcomings of the MB, as discussed in section IV. Due to this, there has been a lot of interest in the hybrid modeling domain. Despite the significant recent progress in the hybrid modeling domain, there are various challenges that throttle down the full-fledged growth of hybrid modeling. They must be addressed in order to exploit the full capacity of hybrid models. In the following subsections, we discuss a few relevant directions for future work in the hybrid modeling domain.

A. HYBRID MODELS: MODEL SELECTION

Model selection is crucial in the hybrid modeling domain. Model selection in physics-based and machine learning domains are quite intuitive. But, not so in the case of hybrid models. Appropriate MBs are selected based on the system, its dynamics, and associated parameters. The data generated by a system can be broadly classified into spatial data, sequential data, or a combination of them. Several MBs model them individually or collectively. Among the deep learning techniques for data processing, CNN and LSTM architectures are prominent. The CNNs are effective at processing spatial data, and LSTMs are efficient at processing sequential data. Their combination is used for processing a combination of spatial and sequential data. Autoencoders [6] are used to extract the abstract features of a dataset in an unsupervised manner. So, there is a clear intuition for model selection in both the physics and the deep learning space. However, there is no clear guideline for model selection in the hybrid modeling space. A well-defined set of guidelines for selecting hybrid models will improve the performance of hybrid models in all aspects and broaden their scope of applications.

B. BENCHMARK PROBLEMS IN HYBRID DOMAIN

A set of benchmark problems eases the evaluation and comparison of different algorithms. Benchmark problems are essential for the growth of every domain. Currently, the ImageNet database [112], [113] and the inverted pendulum [254], [255] serve as the benchmark problems for object detection and reinforcement learning tasks respectively. But, no benchmark CPS problems or dataset exists for evaluating and comparing the hybrid models. In addition to that, the benchmark problems are specific to each application and evolve with the success of the state of the art algorithms. The benchmark problems in object detection and recognition evolved from CalTech 101 [114] to CIFAR-10 [115] to ImageNet [112], [113]. In general, the growth of the state of the

art techniques can be directly inferred from the complexity of the benchmark problems. So, the benchmark problems for hybrid modeling in CPS shall also evolve with the expertise. Also, as CPS also has diverse applications, it may require the formulation of unique benchmark problems for each domain.

C. REGRESSION PROBLEMS WITH LOW AMOUNT OF DATA

Supervised learning algorithms try to “learn” accurate models that can predict the value of the dependent attribute (output) from the attribute variables (inputs). Regression is one of the two main groups of supervised learning problems (classification being the other) in which the output variable is a real or continuous value. Regression problems manifest in a multitude of CPS domain problems. Modifying, training, and tuning hybrid models for a regression problem is non-trivial in comparison to classification problems.

Additionally, in between the spectrum from full physics to no physics and from zero data to big data, there lies an area in the hybrid modeling domain that has limited data and limited physics. A large class of CPS modeling tasks belongs to the limited data and limited physics realm. Training accurate hybrid models for regression problems with a low amount of data is a fertile field for future research. The prior works [133], [200], [215], [224], [225] has shown that hybrid models considerably outperformed both MB and ML in terms of interpolability and extrapolability in a variety of applications with limited data. However, the application problems in these existing work are limited to simple systems. They have not been exclusively tested in CPS applications. Hence, an exclusive and comprehensive evaluation of hybrid models against CPS with limited data is required.

D. LEARNING WITH LESS LABELS

With enough data, one can build accurate hybrid or data-driven models. Accurate hybrid models currently require lots of labeled data. The commercial world (image, video, and text analytics) have curated large sets of labeled data for training models through cheap but effective crowdsourcing methods. Unfortunately, crowdsourcing techniques are not amenable for creating datasets for the CPS domain. Alternate means of data labeling can result in a 100x higher cost and longer time to label. Developing novel hybrid models that learn with fewer labels opens up a new avenue of research and practical applications. Specifically, semi-supervised hybrid learning approaches that use less amount of labeled data along with a huge amount of unlabeled data is an interesting avenue of research in the CPS hybrid modeling domain.

Moreover, as CPS finds applications in diverse domains, labeling may require domain-specific expertise. So, utilizing techniques like active learning [256] and semi-supervised learning [257] for hybrid modeling in general, and CPS in particular, is a fruitful avenue of research. Hybrid learning techniques are yet to use concepts from active and semi-supervised learning. Additionally, a set of metrics for unsupervised learning [258] in CPS should also be developed.

The success of unsupervised learning in CPS hybrid modeling can truly make the CPS ubiquitous.

E. INTELLIGENT COLLECTION OF DATASETS FOR HYBRID MODELS

Data collection is an expensive process. Moreover, data is always sparse and can never completely represent the state space. So, data shall be collected in an intelligent manner to capture the global behavior of the system. As the model is trained on the set of acquired data, the model tends to improve its interpolability (MSE_{Int}). However, the performance of the model against new data, i.e., extrapolability (MSE_{Ext}) is not guaranteed. The extrapolability of the model will be at par with the interpolability if the new test data is in the vicinity of the space of the training data and vice versa (Figure 8(b)). Hybrid models are capable of learning the complete model of the system. Hence, to develop a complete model of the system, new data shall be collected from the regions of the system's space that are far from the training set, in which the model is likely to fail. The MSE_{Ext} will be high. The system shall be simulated to capture exceptional cases like system failure and abrupt behavior. Such data acquisition will be instrumental for systems where the consequences of failure can be catastrophic. The model shall be trained on such data for reducing its bias and until its accuracy saturates. Then, again a new region shall be sought. And the process shall be repeated. To conclude, the acquisition of new data and training of the model shall be conducted in a non-cooperative game-theoretic approach for influencing the mean square error. Additional techniques for data collection, as described in [259] can also be used. The accuracy of the model can be expected to improve in steps where each step corresponds to a new region of data that is not in the proximity of the trained data.

F. GEOMETRIC MACHINE LEARNING IN CYBER-PHYSICAL SYSTEMS

CPS has opened up the gates for the flood of big data in every imaginable domain. Big data is accompanied by high dimensional input data [111]. When the dimensionality of the data is very high, the useful data is actually embedded in a low dimensional manifold. Learning such manifolds comes under the purview of geometric deep learning. Recent work in the CPS [260], [261] has forayed into geometric deep learning, where the data is non-Euclidean. In a non-Euclidean setting, the properties of Euclidean geometry like translation invariance and stability against local deformations [262] are no more valid globally. They are only valid locally. So, the CNNs, which leverage these two properties for processing spatial data, cannot be used anymore. That invites novel concepts from geometric deep learning into the cyber-physical space. Among the geometric deep learning techniques, structured graph CNNs [261] and spectral clustering [263] techniques have already been utilized for CPS. Other techniques like spectral CNNs [264], [265], graph CNNs [266], [267], geodesic CNNs [268], manifold learning should be explored

for CPS applications. Geometric deep learning techniques can be instrumental for processing big data [269] in a cyber-physical setup.

G. IMBALANCED DATA AND DATA AT TAIL

The data, in general, is likely to be imbalanced, i.e., not all the classes will have even representation. Its impact becomes critical in security applications, disease classifications, fraud and fault detection, etc. In most cases, misclassifying the minority class (e.g., disease or a perpetrator), which does not have enough data for allowing the algorithm to learn, is probable. The impact of misclassification in case of disease prediction or security applications can be fatal or catastrophic. The techniques for controlling the adverse impact of class balance manipulates the data or the learning algorithm or both [270]. However, their performance at mitigating the impact of class imbalance is not as expected and hence debatable [270]–[272]. Additionally, data at the tail-end, for example, anomalies in CPS system behaviors are hard to curate. An anomalous data sample can be considered to be generated by distribution (tail-end) that is significantly different from the normal behavior distribution. In general, anomalies are too expensive, time-consuming, and sometimes even dangerous to record. Therefore anomalies are not observed abundantly on real CPS systems. We still have a long way to go for correctly addressing imbalanced data and data at the tail in the context of CPS hybrid modeling.

VIII. CONCLUSION

This is a diligent attempt at disseminating the collective intelligence of hybrid models in an organized manner. We have classified the hybrid models into physics-based preprocessing, physics-based network architectures, physics-based regularization, and miscellaneous categories based on the way the MB is brought into the hybrid architecture. The manner in which their synergy can help eliminate the challenges posed by big data, probabilistic nature of CPS, environmental chaos, and limits of determinism to the existing modeling techniques of CPS is also elucidated. In addition to this, we have illustrated how the MB can eradicate the limitations of ML models arising due to the time resolution of the data acquisition system. Overall, the combination is a win-win situation. Moreover, we proposed five metrics for all-round performance evaluation of a hybrid CPS model. The discussed metrics go beyond the conventional metrics of accuracy within the training dataset. Finally, the challenges faced by hybrid models that deter the growth of hybrid models in the CPS domain are briefed. We hope the current work to become a stepping stone at the organized dispersal and growth of CPS hybrid models.

REFERENCES

- [1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [2] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.

- [3] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," *Handbook Brain Theory Neural Netw.*, vol. 3361, no. 10, p. 1995, 1995.
- [4] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," in *Proc. ICANN*, 1999, pp. 850–855.
- [5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
- [6] C. Doersch, "Tutorial on variational autoencoders," 2016, *arXiv:1606.05908*. [Online]. Available: <http://arxiv.org/abs/1606.05908>
- [7] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
- [8] S. Skansi, *Introduction to Deep Learning From Logical Calculus to Artificial Intelligence*. Springer, 2018.
- [9] E. A. Lee and S. A. Seshia, *Introduction to Embedded Systems: A Cyber-Physical Systems Approach*. Cambridge, MA, USA: MIT Press, 2016.
- [10] R. Baheti and H. Gill, "Cyber-physical systems," *Impact Control Technol.*, vol. 12, pp. 161–166, Mar. 2011.
- [11] R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, "Cyber-physical systems: The next computing revolution," in *Proc. 47th Design Autom. Conf. (DAC)*, 2010, pp. 731–736.
- [12] J. Lee, B. Bagheri, and H.-A. Kao, "A cyber-physical systems architecture for industry 4.0-based manufacturing systems," *Manuf. Lett.*, vol. 3, pp. 18–23, Jan. 2015.
- [13] B. Vogel-Heuser, J. Lee, and P. Leitão, "Agents enabling cyber-physical production systems," *Automatisierungstechnik*, vol. 63, no. 10, pp. 777–789, Jan. 2015.
- [14] A. Karpatne, G. Atluri, J. H. Faghmous, M. Steinbach, A. Banerjee, A. Ganguly, S. Shekhar, N. Samatova, and V. Kumar, "Theory-guided data science: A new paradigm for scientific discovery from data," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 10, pp. 2318–2331, Oct. 2017.
- [15] I. Lee, *Cyber Physical Systems: The Next Computing Revolution*. Accessed: Spring 2010. [Online]. Available: <https://www.seas.upenn.edu/~lee/10cis541/lects/lec-CPS-1x2.pdf>
- [16] S. A. Haque, S. M. Aziz, and M. Rahman, "Review of cyber-physical system in healthcare," *Int. J. Distrib. Sensor Netw.*, vol. 10, no. 4, Apr. 2014, Art. no. 217415.
- [17] H. A. Muller, "The rise of intelligent cyber-physical systems," *Computer*, vol. 50, no. 12, pp. 7–9, Dec. 2017. [Online]. Available: <https://www.computer.org/csdl/magazine/co/2017/12/mco2017120007/13rUytf44K>
- [18] P. Goldstein, (2018). *How do Feds Use and Secure Cyber-Physical Systems?* [Online]. Available: <https://fedtechmagazine.com/article/2018/01/how-do-feds-use-and-secure-cyber-physical-systems>
- [19] R. Sabella, (2018). *Cyber Physical Systems for Industry 4.0*. [Online]. Available: <https://www.ericsson.com/en/blog/2018/10/cyber-physical-systems-for-industry-4-0>
- [20] J. Shi, J. Wan, H. Yan, and H. Suo, "A survey of cyber-physical systems," in *Proc. Int. Conf. Wireless Commun. Signal Process. (WCSP)*, 2011, pp. 1–6.
- [21] H. Gill, (2006). *NSF Perspective and Status on Cyber-Physical Systems*. Accessed: 2015. [Online]. Available: <http://varma.ece.cmu.edu/CPS/Presentations/gill.pdf>
- [22] Networking, I. T. Research, (2015). *Cyber Physical Systems*. [Online]. Available: https://www.nitrd.gov/nitrdgroups/images/6/6a/ cyber_Physical_Systems_%28CPS%29_Vision_Statement.pdf
- [23] F. Mehdipour, K. C. Nunna, and K. J. Murakami, "A smart cyber-physical systems-based solution for pest control (work in progress)," in *Proc. IEEE Int. Conf. Green Comput. Commun. IEEE Internet Things IEEE Cyber, Phys. Social Comput.*, Aug. 2013, pp. 1248–1253.
- [24] F. Mehdipour, "Smart field monitoring: An application of cyber-physical systems in agriculture (work in progress)," in *Proc. IIAI 3rd Int. Conf. Adv. Appl. Inform.*, Aug. 2014, pp. 181–184.
- [25] S. I. Caramihai and I. Dumitrache, "Agricultural enterprise as a complex system: A cyber physical systems approach," in *Proc. 20th Int. Conf. Control Syst. Comput. Sci.*, May 2015, pp. 659–664.
- [26] C.-R. Rad, O. Hancu, I.-A. Takacs, and G. Olteanu, "Smart monitoring of potato crop: A cyber-physical system architecture model in the field of precision agriculture," *Agric. Sci. Procedia*, vol. 6, pp. 73–79, Jan. 2015.
- [27] C.-U. Lei, H.-N. Liang, and K. L. Man, "Building a smart laboratory environment at a university via a cyber-physical system," in *Proc. IEEE Int. Conf. Teaching, Assessment Learn. Eng. (TALE)*, Aug. 2013, pp. 243–247.
- [28] C.-U. Lei, K. Wan, and K. L. Man, "Developing a smart learning environment in universities via cyber-physical systems," *Procedia Comput. Sci.*, vol. 17, pp. 583–585, Jan. 2013.
- [29] V. Riquebourg, D. Menga, D. Durand, B. Marhic, L. Delahoché, and C. Loge, "The smart home concept: Our immediate future," in *Proc. 1st IEEE Int. Conf. E-Learn. Ind. Electron.*, Dec. 2006, pp. 23–28.
- [30] F. Basile, P. Chiacchio, J. Coppola, and D. Gerbasio, "Automated warehouse systems: A cyber-physical system perspective," in *Proc. IEEE 20th Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2015, pp. 1–4.
- [31] Q. Shafi, "Cyber physical systems security: A brief survey," in *Proc. 12th Int. Conf. Comput. Sci. Appl.*, Jun. 2012, pp. 146–150.
- [32] P. Dong, Y. Han, X. Guo, and F. Xie, "A systematic review of studies on cyber physical system security," *Int. J. Secur. Its Appl.*, vol. 9, no. 1, pp. 155–164, Jan. 2015.
- [33] S. Sridhar, A. Hahn, and M. Govindarasu, "Cyber-physical system security for the electric power grid," *Proc. IEEE*, vol. 100, no. 1, pp. 210–224, Jan. 2012.
- [34] Y. Li, J. Wu, and S. Li, "State estimation for distributed cyber-physical power systems under data attacks," *Int. J. Model., Identificat. Control*, vol. 26, no. 4, pp. 317–323, 2016.
- [35] A. P. Hamins, N. P. Bryner, A. W. Jones, and G. H. Koepke, "Research roadmap for smart fire fighting," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. 1191, 2015, doi: [10.6028/NIST.SP.1191](https://doi.org/10.6028/NIST.SP.1191).
- [36] V. V. Estrela, J. Hemanth, O. Saotome, E. G. Grata, and D. R. Izario, "Emergency response cyber-physical system for flood prevention with sustainable electronics," in *Proc. Brazilian Technol. Symp.* Springer, 2017, pp. 319–328.
- [37] E. Gelenbe and F.-J. Wu, "Future research on cyber-physical emergency management systems," *Future Internet*, vol. 5, no. 3, pp. 336–354, 2013.
- [38] V. Estrela, O. Saotome, H. Loschi, J. Hemanth, W. Farfan, J. Aroma, C. Saravanan, and E. Grata, "Emergency response cyber-physical framework for landslide avoidance with sustainable electronics," *Technologies*, vol. 6, no. 2, p. 42, 2018.
- [39] Y.-T. Li, M. Jacob, G. Akingba, and J. P. Wachs, "A cyber-physical management system for delivering and monitoring surgical instruments in the OR," *Surgical Innov.*, vol. 20, no. 4, pp. 377–384, Aug. 2013.
- [40] D. Sonntag, S. Zillner, S. Chakraborty, A. Lorincz, E. Strommer, and L. Serafini, "The medical cyber-physical system for activity at EIT: A look under the hood," in *Proc. IEEE 27th Int. Symp. Computer-Based Med. Syst.*, May 2014, pp. 351–356.
- [41] C. G. Cassandras, "Smart cities as cyber-physical social systems," *Engineering*, vol. 2, no. 2, pp. 156–158, Jun. 2016.
- [42] D. Jin, C. Hannon, Z. Li, P. Cortes, S. Ramaraju, P. Burgess, N. Buch, and M. Shahidepour, "Smart street lighting system: A platform for innovative smart city applications and a new frontier for cyber-security," *Electr. J.*, vol. 29, no. 10, pp. 28–35, Dec. 2016.
- [43] J. Wan, H. Yan, D. Li, K. Zhou, and L. Zeng, "Cyber-physical systems for optimal energy management scheme of autonomous electric vehicle," *Comput. J.*, vol. 56, no. 8, pp. 947–956, Aug. 2013.
- [44] G. Xiong, F. Zhu, X. Liu, X. Dong, W. Huang, S. Chen, and K. Zhao, "Cyber-physical-social system in intelligent transportation," *IEEE/CAA J. Automatica Sinica*, vol. 2, no. 3, pp. 320–333, Jul. 2015.
- [45] T. Sanislav, G. Mois, S. Folea, L. Miclea, G. Gambardella, and P. Prinetto, "A cloud-based cyber-physical system for environmental monitoring," in *Proc. 3rd Medit. Conf. Embedded Comput. (MECO)*, Jun. 2014, pp. 6–9.
- [46] C. Liu and P. Jiang, "A cyber-physical system architecture in shop floor for intelligent manufacturing," *Procedia Cirp*, vol. 56, pp. 372–377, 2016.
- [47] B. Bagheri, S. Yang, H.-A. Kao, and J. Lee, "Cyber-physical systems architecture for self-aware machines in industry 4.0 environment," *IFAC-PapersOnLine*, vol. 48, no. 3, pp. 1622–1627, 2015.
- [48] R. F. Babiceanu and R. Seker, "Big data and virtualization for manufacturing cyber-physical systems: A survey of the current status and future outlook," *Comput. Ind.*, vol. 81, pp. 128–137, Sep. 2016.
- [49] J. Lee, H. D. Ardakani, S. Yang, and B. Bagheri, "Industrial big data analytics and cyber-physical systems for future maintenance & service innovation," *Procedia Cirp*, vol. 38, pp. 3–7, 2015.
- [50] H. Chen, "Applications of cyber-physical system: A literature review," *J. Ind. Integr. Manage.*, vol. 2, no. 3, Sep. 2017, Art. no. 1750012.
- [51] S. K. Khaitan and J. D. McCalley, "Cyber physical system approach for design of power grids: A survey," in *Proc. IEEE Power Energy Soc. Gen. Meeting*, 2013, pp. 1–5.

- [52] C. Yin, Z. Xiong, H. Chen, J. Wang, D. Cooper, and B. David, "A literature survey on smart cities," *Sci. China Inf. Sci.*, vol. 58, no. 10, pp. 1–18, 2015.
- [53] D. Jia, K. Lu, J. Wang, X. Zhang, and X. Shen, "A survey on platoon-based vehicular cyber-physical systems," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 263–284, 1st Quart., 2016.
- [54] T. H. Davenport and J. Kirby. (2015). *Beyond Automation*. [Online]. Available: <https://ibm.ent.box.com/s/5fhhabp83jafccmsbmyvlnq2ocbw188>
- [55] G. Ducard, "Modeling and analysis of dynamic systems," *Inst. Dyn. Syst. Control*, no. 2016, pp. 1–21, 2016. [Online]. Available: https://ethz.ch/content/dam/ethz/special-interest/mavt/dynamic-systems-n-control/idsc-dam/Lectures/System-Modeling/Slides_HS16/Lecture4.pdf
- [56] B. Huyck, J. De Brabanter, J. Van Impe, and B. De Moor, "Identification and modeling of a dynamical system," in *Proc. 3rd Eur. Conf. Modern Inf. Commun. Technol. (ECUMICT)*, Jan. 2008, pp. 1–12. [Online]. Available: <ftp://ftp.esat.kuleuven.be/stadius/bhuyck/Ecumict2008.pdf>
- [57] J. Ledin, "Modeling dynamic systems," *Embedded Syst. Program.*, p. 85, Aug. 2000.
- [58] S. Moura. (2018). *Chapter 1: Modeling and Systems Analysis*. [Online]. Available: <https://ecal.berkeley.edu/files/ce295/CH01-ModelingSystems.pdf>
- [59] A. T. Bahill and F. Szidarovszky, "Comparison of dynamic system modeling methods," *Syst. Eng.*, vol. 12, no. 3, pp. 183–200, Jun. 2009.
- [60] D. Baraff, "Physically based modeling: Rigid body simulation," *SIGGRAPH Course Notes, ACM*, vol. 2, no. 1, pp. 1–2, 2001.
- [61] O. Badmash, K. Eveker, and C. Nett, "Control-oriented high-frequency turbomachinery modeling. i-theoretical foundations," in *Proc. 28th Joint Propuls. Conf. Exhibit*, 1993, p. 3314.
- [62] H. Hjalmarsson and C. R. Rojas. Physical and Data Driven Modelling for Control With Applications From the Paper Industry. Nordic Process Control Workshop Tutorial, 2016. [Online]. Available: https://www.kth.se/polopoly_fs/1.667999.1550155815/tutorial_cristian.pdf
- [63] P. P. Chu, *RTL Hardware Design Using VHDL: Coding for Efficiency, Portability, and Scalability*. Hoboken, NJ, USA: Wiley, 2006.
- [64] V. A. Pedroni, *Finite State Machines in Hardware: Theory and Design (With VHDL and SystemVerilog)*. Cambridge, MA, USA: MIT Press, 2013.
- [65] D. Lee and M. Yannakakis, "Principles and methods of testing finite state machines—A survey," *Proc. IEEE*, vol. 84, no. 8, pp. 1090–1123, 1996.
- [66] G. H. Mealy, "A method for synthesizing sequential circuits," *Bell Syst. Tech. J.*, vol. 34, no. 5, pp. 1045–1079, Sep. 1955.
- [67] E. F. Moore, "Gedanken-experiments on sequential machines," *Automata Stud.*, vol. 34, pp. 129–153, Dec. 1956.
- [68] J. Lin, S. Sedigh, and A. Miller, "Modeling cyber-physical systems with semantic agents," in *Proc. IEEE 34th Annu. Comput. Softw. Appl. Conf. Workshops*, Jul. 2010, pp. 13–18.
- [69] J. Lin, S. Sedigh, and A. Miller, "A semantic agent framework for cyber-physical systems," in *Semantic Agent Systems*. Springer, 2011, pp. 189–213.
- [70] P. Leitão, S. Karnouskos, L. Ribeiro, J. Lee, T. Strasser, and A. W. Colombo, "Smart agents in industrial cyber-physical systems," *Proc. IEEE*, vol. 104, no. 5, pp. 1086–1101, May 2016.
- [71] E. Bonabeau, "Agent-based modeling: Methods and techniques for simulating human systems," *Proc. Nat. Acad. Sci. USA*, vol. 99, no. 3, pp. 7280–7287, May 2002.
- [72] B. Subagdja, "Agent-based modeling for developing pervasive persuasive systems," in *Proc. Int. Conf. Adv. Comput. Sci. Inf. Syst. (ICACSIS)*, Oct. 2015, pp. 23–28.
- [73] A. Hector and V. L. Narasimhan, "A new classification scheme for software agents," in *Proc. 3rd Int. Conf. Inf. Technol. Appl. (ICITA)*, vol. 1, 2005, pp. 191–196.
- [74] A. Aliyuda, "Towards the design of cyber-physical system via multi-agent system technology," *Int. J. Sci. Eng. Res.*, vol. 7, no. 10, pp. 155–161, 2016.
- [75] P. Pederson, D. Dudenhofer, S. Hartley, and M. Permann, "Critical infrastructure interdependency modeling: A survey of us and international research," *Idaho Nat. Lab.*, vol. 25, p. 27, Aug. 2006.
- [76] A. L. Juloski, S. Weiland, and W. P. M. H. Heemels, "A Bayesian approach to identification of hybrid systems," *IEEE Trans. Autom. Control*, vol. 50, no. 10, pp. 1520–1533, Oct. 2005.
- [77] R. Vidal, S. Soatto, Y. Ma, and S. Sastry, "An algebraic geometric approach to the identification of a class of linear hybrid systems," in *Proc. 42nd IEEE Int. Conf. Decis. Control*, vol. 1, Dec. 2003, pp. 167–172.
- [78] A. Bemporad, A. Garulli, S. Paoletti, and A. Vicino, "A bounded-error approach to piecewise affine system identification," *IEEE Trans. Autom. Control*, vol. 50, no. 10, pp. 1567–1580, Oct. 2005.
- [79] N. Ozay, M. Sznajder, C. M. Lagoa, and O. I. Camps, "A sparsification approach to set membership identification of switched affine systems," *IEEE Trans. Autom. Control*, vol. 57, no. 3, pp. 634–648, Mar. 2012.
- [80] Y. Yuan, X. Tang, W. Zhou, W. Pan, X. Li, H.-T. Zhang, H. Ding, and J. Goncalves, "Data driven discovery of cyber physical systems," *Nature Commun.*, vol. 10, no. 1, p. 4894, Dec. 2019, doi: [10.1038/s41467-019-12490-1](https://doi.org/10.1038/s41467-019-12490-1).
- [81] V. Breschi, D. Piga, and A. Bemporad, "Piecewise affine regression via recursive multiple least squares and multicategory discrimination," *Automatica*, vol. 73, pp. 155–162, Nov. 2016.
- [82] D. L. Ly and H. Lipson, "Learning symbolic representations of hybrid dynamical systems," *J. Mach. Learn. Res.*, vol. 13, pp. 3585–3618, Dec. 2012.
- [83] G. Pillonetto, "A new kernel-based approach to hybrid system identification," *Automatica*, vol. 70, pp. 21–31, Aug. 2016.
- [84] S. Paoletti, A. L. Juloski, G. Ferrari-Trecate, and R. Vidal, "Identification of hybrid systems a tutorial," *Eur. J. Control*, vol. 13, nos. 2–3, pp. 242–260, Jan. 2007.
- [85] H. Nakada, K. Takaba, and T. Katayama, "Identification of piecewise affine systems based on statistical clustering technique," *Automatica*, vol. 41, no. 5, pp. 905–913, May 2005.
- [86] R. H. R. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, and H. S. Seung, "Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit," *Nature*, vol. 405, no. 6789, pp. 947–951, Jun. 2000.
- [87] G. E. Dahl, T. N. Sainath, and G. E. Hinton, "Improving deep neural networks for LVCSR using rectified linear units and dropout," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 8609–8613.
- [88] M. D. Zeiler, M. Ranzato, R. Monga, M. Mao, K. Yang, Q. V. Le, P. Nguyen, A. Senior, V. Vanhoucke, J. Dean, and G. E. Hinton, "On rectified linear units for speech processing," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 3517–3521.
- [89] J. Han and C. Moraga, "The influence of the sigmoid function parameters on the speed of backpropagation learning," in *Proc. Int. Workshop Artif. Neural Netw.* Springer, 1995, pp. 195–201.
- [90] T. Drossi, A. Donzé, and S. A. Seshia, "Compositional falsification of cyber-physical systems with machine learning components," *J. Automated Reasoning*, vol. 63, no. 4, pp. 1031–1053, Dec. 2019.
- [91] C. S. Wickramasinghe, D. L. Marino, K. Amarasinghe, and M. Manic, "Generalization of deep learning for cyber-physical system security: A survey," in *Proc. 44th Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Oct. 2018, pp. 745–751.
- [92] A. Platzer, "The logical path to autonomous cyber-physical systems," in *Proc. Int. Conf. Quant. Eval. Syst.* Springer, 2019, pp. 25–33.
- [93] P. Derler, E. A. Lee, and A. S. Vincentelli, "Modeling cyber-physical systems," *Proc. IEEE*, vol. 100, no. 1, pp. 13–28, Jan. 2012.
- [94] P. K. Menon, "Mathematical models of dynamical systems for control system design," in *Proc. IEEE Control Syst. Soc. Seminar*, 2011, pp. 1–14. [Online]. Available: <https://ewh.ieee.org/r6/scv/css/archive/022011PKMenon.pdf>
- [95] E. A. Lee, "Fundamental limits of cyber-physical systems modeling," *ACM Trans. Cyber-Phys. Syst.*, vol. 1, no. 1, pp. 1–26, Feb. 2017.
- [96] A. D. Ames, A. Abate, and S. Sastry, "Sufficient conditions for the existence of zeno behavior," in *Proc. 44th IEEE Conf. Decis. Control*, 2005, pp. 696–701.
- [97] L. Thiele and P. Kumar, "Can real-time systems be chaotic?" in *Proc. Int. Conf. Embedded Softw. (EMSOFT)*, Oct. 2015, pp. 21–30.
- [98] Z. Liu, D.-S. Yang, D. Wen, W.-M. Zhang, and W. Mao, "Cyber-physical-social systems for command and control," *IEEE Intell. Syst.*, vol. 26, no. 4, pp. 92–96, Jul. 2011.
- [99] J. Zeng, L. T. Yang, M. Lin, H. Ning, and J. Ma, "A survey: Cyber-physical-social systems and their system-level design methodology," *Future Gener. Comput. Syst.*, vol. 105, pp. 1028–1042, Apr. 2020.
- [100] J. J. Zhang, F.-Y. Wang, X. Wang, G. Xiong, F. Zhu, Y. Lv, J. Hou, S. Han, Y. Yuan, Q. Lu, and Y. Lee, "Cyber-Physical-Social systems: The state of the art and perspectives," *IEEE Trans. Comput. Social Syst.*, vol. 5, no. 3, pp. 829–840, Sep. 2018.
- [101] F.-Y. Wang, X. Wang, L. Li, and L. Li, "Steps toward parallel intelligence," *IEEE/CAA J. Automatica Sinica*, vol. 3, no. 4, pp. 345–348, Oct. 2016.

- [102] W. E. Wilkins, "The concept of a self-fulfilling prophecy," *Sociol. Edu.*, vol. 49, no. 2, pp. 175–183, Apr. 1976.
- [103] D. Krishna, "'The self-fulfilling prophecy' and the nature of society," *Amer. Sociol. Rev.*, vol. 36, no. 6, pp. 1104–1107, 1971.
- [104] P. Czop, G. Kost, D. Sławik, and G. Wszolek, "Formulation and identification of first-principle data-driven models," *J. Achievements Mater. Manuf. Eng.*, vol. 44, no. 2, pp. 179–186, 2011.
- [105] A. Gandomi and M. Haider, "Beyond the hype: Big data concepts, methods, and analytics," *Int. J. Inf. Manage.*, vol. 35, no. 2, pp. 137–144, Apr. 2015.
- [106] M. A.-U.-D. Khan, M. F. Uddin, and N. Gupta, "Seven V's of big data understanding big data to extract value," in *Proc. Zone 1 Conf. Amer. Soc. Eng. Edu.*, Apr. 2014, pp. 1–5.
- [107] Y. Demchenko, P. Grosso, C. de Laat, and P. Membrey, "Addressing big data issues in scientific data infrastructure," in *Proc. Int. Conf. Collaboration Technol. Syst. (CTS)*, May 2013, pp. 48–55.
- [108] R. Patgiri and A. Ahmed, "Big data: The V's of the game changer paradigm," in *Proc. IEEE 18th Int. Conf. High Perform. Comput. Commun., IEEE 14th Int. Conf. Smart City, IEEE 2nd Int. Conf. Data Sci. Syst. (HPCC/SmartCity/DSS)*, Dec. 2016, pp. 17–24.
- [109] D. Laney, "3D data management: Controlling data volume, velocity and variety," *META Group Res. note*, vol. 6, no. 70, p. 1, 2001.
- [110] R. Kitchin and G. McArdle, "What makes big data, big data? Exploring the ontological characteristics of 26 datasets," *Big Data Soc.*, vol. 3, no. 1, Jan. 2016, Art. no. 205395171663113.
- [111] V. Gavrishchaka, O. Senyukova, and M. Koepke, "Synergy of physics-based reasoning and machine learning in biomedical applications: Towards unlimited deep learning with limited data," *Adv. Phys.*, X, vol. 4, no. 1, Jan. 2019, Art. no. 1582361.
- [112] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
- [113] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.
- [114] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories," in *Proc. Conf. Comput. Vis. Pattern Recognit. Workshop*, 2004, p. 178.
- [115] A. Krizhevsky and G. Hinton, "Convolutional deep belief networks on CIFAR-10," *Unpublished Manuscript*, vol. 40, no. 7, pp. 1–9, 2010.
- [116] K. Boman. (2014). *Marriage of Analytics, Engineering First Principles Needed in Oil and Gas*. [Online]. Available: https://www.rigzone.com/news/oil_gas/a/135690/Marriage_of_Analytics_Engineering_First_Principles_Needed_in_Oil_and_Gas/
- [117] C. Metz. (2016). *How Google's AI Viewed the Move no Human Could Understand*. [Online]. Available: https://www.wired.com/2016/03/googles-ai-viewed-move-no-human-understand_d/
- [118] G. E. Box, *All Models Are Wrong, But Some Are Useful*, vol. 202. Robustness in Statistics, 1979, p. 549. [Online]. Available: <https://www.actuaries.org.uk/system/files/documents/pdf/d5-presentation.pdf>
- [119] A. Saltelli and S. Funtowicz, "When all models are wrong," *Issues Sci. Technol.*, vol. 30, no. 2, pp. 79–85, 2014.
- [120] D. Ghosh, E. Hermonat, P. Mhaskar, S. Snowling, and R. Goel, "Hybrid modeling approach integrating first-principles models with subspace identification," *Ind. Eng. Chem. Res.*, vol. 58, no. 30, pp. 13533–13543, Jul. 2019.
- [121] C. C. Gülçehre and Y. Bengio, "Knowledge matters: Importance of prior information for optimization," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 226–257, 2016.
- [122] P. Niyogi, F. Girosi, and T. Poggio, "Incorporating prior information in machine learning by creating virtual examples," *Proc. IEEE*, vol. 86, no. 11, pp. 2196–2209, 1998.
- [123] A. Karpatne, W. Watkins, J. Read, and V. Kumar, "Physics-guided neural networks (PGNN): An application in lake temperature modeling," 2017, *arXiv:1710.11431*. [Online]. Available: <http://arxiv.org/abs/1710.11431>
- [124] X. Jia, A. Karpatne, J. Willard, M. Steinbach, J. Read, P. C. Hanson, H. A. Dugan, and V. Kumar, "Physics guided recurrent neural networks for modeling dynamical systems: Application to monitoring water temperature and quality in lakes," 2018, *arXiv:1810.02880*. [Online]. Available: <http://arxiv.org/abs/1810.02880>
- [125] X. Jia, J. Willard, A. Karpatne, J. Read, J. Zwart, M. Steinbach, and V. Kumar, "Physics guided RNNs for modeling dynamical systems: A case study in simulating lake temperature profiles," in *Proc. SIAM Int. Conf. Data Mining*, 2019, pp. 558–566.
- [126] Y. Yang, L. Sanchez, H. Zhang, A. Roeder, J. Bowlan, J. Crochet, C. Farrar, and D. Mascareñas, "Estimation of full-field, full-order experimental modal model of cable vibration from digital video measurements with physics-guided unsupervised machine learning and computer vision," *Struct. Control Health Monitor.*, vol. 26, no. 6, p. e2358, Jun. 2019.
- [127] Z. Ren, S. Tian, T. Heumueller, E. Birgersson, F. Lin, A. Aberle, S. Sun, I. M. Peters, R. Stangl, C. J. Brabec, T. Buonassisi, F. Oviedo, H. Xue, M. Thway, K. Zhang, N. Li, J. D. Perea, M. Layurova, and Y. Wang, "Physics-guided characterization and optimization of solar cells using surrogate machine learning model," in *Proc. IEEE 46th Photovoltaic Specialists Conf. (PVSC)*, Jun. 2019, pp. 3054–3058.
- [128] Y. Ba, G. Zhao, and A. Kadambi, "Blending diverse physical priors with neural networks," 2019, *arXiv:1910.00201*. [Online]. Available: <http://arxiv.org/abs/1910.00201>
- [129] A. Zeng, S. Song, J. Lee, A. Rodriguez, and T. Funkhouser, "Tossing-Bot: Learning to throw arbitrary objects with residual physics," 2019, *arXiv:1903.11239*. [Online]. Available: <http://arxiv.org/abs/1903.11239>
- [130] J. Nagoro Kani and A. H. Elsheikh, "DR-RNN: A deep residual recurrent neural network for model reduction," 2017, *arXiv:1709.00939*. [Online]. Available: <http://arxiv.org/abs/1709.00939>
- [131] A. Ajay, J. Wu, N. Fazeli, M. Bauza, L. P. Kaelbling, J. B. Tenenbaum, and A. Rodriguez, "Augmenting physical simulators with stochastic neural networks: Case study of planar pushing and bouncing," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 3066–3073.
- [132] Y. Ba, R. Chen, Y. Wang, L. Yan, B. Shi, and A. Kadambi, "Physics-based neural networks for shape from polarization," 2019, *arXiv:1903.10210*. [Online]. Available: <http://arxiv.org/abs/1903.10210>
- [133] S. K. Singh, R. Yang, A. Behjat, R. Rai, S. Chowdhury, and I. Matei, "PI-LSTM: Physics-infused long short-term memory network," in *Proc. 18th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2019.
- [134] N. Wagner and J. M. Rondinelli, "Theory-guided machine learning in materials science," *Frontiers Mater.*, vol. 3, p. 28, Jun. 2016.
- [135] Y. Long, X. She, and S. Mukhopadhyay, "HybridNet: Integrating model-based and data-driven learning to predict evolution of dynamical systems," 2018, *arXiv:1806.07439*. [Online]. Available: <http://arxiv.org/abs/1806.07439>
- [136] H. Hanachi, W. Yu, I. Y. Kim, J. Liu, and C. K. Mechefske, "Hybrid data-driven physics-based model fusion framework for tool wear prediction," *Int. J. Adv. Manuf. Technol.*, vol. 101, nos. 9–12, pp. 2861–2872, Apr. 2019.
- [137] N. Mohajerin and S. L. Waslander, "Multistep prediction of dynamic systems with recurrent neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 11, pp. 3370–3383, Nov. 2019.
- [138] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *J. Comput. Phys.*, vol. 378, pp. 686–707, Feb. 2019.
- [139] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics informed deep learning (Part I): Data-driven solutions of nonlinear partial differential equations," 2017, *arXiv:1711.10561*. [Online]. Available: <http://arxiv.org/abs/1711.10561>
- [140] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics informed deep learning (Part II): Data-driven discovery of nonlinear partial differential equations," 2017, *arXiv:1711.10566*. [Online]. Available: <http://arxiv.org/abs/1711.10566>
- [141] G. Pang, L. Lu, and G. E. Karniadakis, "FPINNs: Fractional physics-informed neural networks," *SIAM J. Scientific Comput.*, vol. 41, no. 4, pp. A2603–A2626, Jan. 2019.
- [142] A. D. Jagtap, K. Kawaguchi, and G. E. Karniadakis, "Adaptive activation functions accelerate convergence in deep and physics-informed neural networks," *J. Comput. Phys.*, vol. 404, Mar. 2020, Art. no. 109136.
- [143] Y. Yang and P. Perdikaris, "Physics-informed deep generative models," 2018, *arXiv:1812.03511*. [Online]. Available: <http://arxiv.org/abs/1812.03511>
- [144] X. Meng, Z. Li, D. Zhang, and G. E. Karniadakis, "PPINN: Parareal physics-informed neural network for time-dependent PDEs," 2019, *arXiv:1909.10145*. [Online]. Available: <http://arxiv.org/abs/1909.10145>

- [145] E. Kharazmi, Z. Zhang, and G. E. Karniadakis, "Variational physics-informed neural networks for solving partial differential equations," 2019, *arXiv:1912.00873*. [Online]. Available: <http://arxiv.org/abs/1912.00873>
- [146] J.-L. Wu, H. Xiao, and E. Paterson, "Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework," *Phys. Rev. Fluids*, vol. 3, no. 7, Jul. 2018, Art. no. 074602.
- [147] Y. Zhu, N. Zabarar, P.-S. Koutsourelakis, and P. Perdikaris, "Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data," *J. Comput. Phys.*, vol. 394, pp. 56–81, Oct. 2019.
- [148] R. Stewart and S. Ermon, "Label-free supervision of neural networks with physics and domain knowledge," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 2576–2582.
- [149] A. Ben-Hur, D. Horn, H. T. Siegelmann, and V. Vapnik, "Support vector clustering," *J. Mach. Learn. Res.*, vol. 2, pp. 125–137, Mar. 2002.
- [150] H. Byun and S.-W. Lee, "Applications of support vector machines for pattern recognition: A survey," in *Proc. Int. Workshop Support Vector Mach.* Springer, 2002, pp. 213–236.
- [151] A. Pradhan, "Support vector machine-a survey," *Int. J. Emerg. Technol. Adv. Eng.*, vol. 2, no. 8, pp. 82–85, 2012.
- [152] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
- [153] A. Verikas, A. Gelzinis, and M. Bacauskiene, "Mining data with random forests: A survey and results of new tests," *Pattern Recognit.*, vol. 44, no. 2, pp. 330–349, Feb. 2011.
- [154] S. García, S. Ramírez-Gallego, J. Luengo, J. M. Benítez, and F. Herrera, "Big data preprocessing: Methods and prospects," *Big Data Anal.*, vol. 1, no. 1, p. 9, 2016.
- [155] O. Faust, Y. Hagiwara, T. J. Hong, O. S. Lih, and U. R. Acharya, "Deep learning for healthcare applications based on physiological signals: A review," *Comput. Methods Programs Biomed.*, vol. 161, pp. 1–13, Jul. 2018.
- [156] Y. Roy, H. Banville, I. Albuquerque, A. Gramfort, T. H. Falk, and J. Faubert, "Deep learning-based electroencephalography analysis: A systematic review," *J. Neural Eng.*, vol. 16, no. 5, 2019, Art. no. 051001.
- [157] W. Zhang, C. Tan, F. Sun, H. Wu, and B. Zhang, "A review of EEG-based brain-computer interface systems design," *Brain Sci. Adv.*, vol. 4, no. 2, pp. 156–167, 2018.
- [158] Z. Zhang, Y. Zhao, X. Liao, W. Shi, K. Li, Q. Zou, and S. Peng, "Deep learning in omics: A survey and guideline," *Briefings Funct. Genomics*, vol. 18, no. 1, pp. 41–57, Feb. 2019.
- [159] F. Zhou, Y. Gao, and C. Wen, "A novel multimode fault classification method based on deep learning," *J. Control Sci. Eng.*, vol. 2017, pp. 1–14, 2017.
- [160] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, and R. X. Gao, "Deep learning and its applications to machine health monitoring," *Mech. Syst. Signal Process.*, vol. 115, pp. 213–237, Jan. 2019.
- [161] Y. Lei, J. Lin, Z. He, and M. J. Zuo, "A review on empirical mode decomposition in fault diagnosis of rotating machinery," *Mech. Syst. Signal Process.*, vol. 35, nos. 1–2, pp. 108–126, Feb. 2013.
- [162] R. Yan, R. X. Gao, and X. Chen, "Wavelets for fault diagnosis of rotary machines: A review with applications," *Signal Process.*, vol. 96, pp. 1–15, Mar. 2014.
- [163] Z. Peng, W. T. Peter, and F. Chu, "A comparison study of improved Hilbert–Huang transform and wavelet transform: Application to fault diagnosis for rolling bearing," *Mech. Syst. signal Process.*, vol. 19, no. 5, pp. 974–988, 2005.
- [164] B. Yang, R. Liu, and X. Chen, "Fault diagnosis for a wind turbine generator bearing via sparse representation and shift-invariant K-SVD," *IEEE Trans. Ind. Informat.*, vol. 13, no. 3, pp. 1321–1331, Jun. 2017.
- [165] R. Liu, B. Yang, E. Zio, and X. Chen, "Artificial intelligence for fault diagnosis of rotating machinery: A review," *Mech. Syst. Signal Process.*, vol. 108, pp. 33–47, Aug. 2018.
- [166] M. Sadoughi and C. Hu, "Physics-based convolutional neural network for fault diagnosis of rolling element bearings," *IEEE Sensors J.*, vol. 19, no. 11, pp. 4181–4192, Jun. 2019.
- [167] M. Sadoughi and C. Hu, "A physics-based deep learning approach for fault diagnosis of rotating machinery," in *Proc. 44th Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Oct. 2018, pp. 5919–5923.
- [168] Y. Qi, W. You, C. Shen, X. Jiang, W. Huang, and Z. Zhu, "Hierarchical diagnosis network based on sparse deep neural networks and its application in bearing fault diagnosis," in *Proc. Prognostics Syst. Health Manage. Conf. (PHM-Harbin)*, Jul. 2017, pp. 1–7.
- [169] J. Hong, Q. Wang, X. Qiu, and H. L. Chan, "Remaining useful life prediction using time-frequency feature and multiple recurrent neural networks," in *Proc. 24th IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2019, pp. 916–923.
- [170] G. Xiao, A. Garg, D. Chen, D. Jiang, W. Shu, and X. Xu, "AHE detection with a hybrid intelligence model in smart healthcare," *IEEE Access*, vol. 7, pp. 37360–37370, 2019.
- [171] C. Venkatesan, P. Karthigaikumar, A. Paul, S. Satheeskumaran, and R. Kumar, "ECG signal preprocessing and SVM classifier-based abnormality detection in remote healthcare applications," *IEEE Access*, vol. 6, pp. 9767–9773, 2018.
- [172] A. Hyvärinen, P. Ramkumar, L. Parkkonen, and R. Hari, "Independent component analysis of short-time Fourier transforms for spontaneous EEG/MEG analysis," *NeuroImage*, vol. 49, no. 1, pp. 257–271, Jan. 2010.
- [173] J. Bedi and D. Toshniwal, "Empirical mode decomposition based deep learning for electricity demand forecasting," *IEEE Access*, vol. 6, pp. 49144–49156, 2018.
- [174] W.-C. Wang, K.-W. Chau, L. Qiu, and Y.-B. Chen, "Improving forecasting accuracy of medium and long-term runoff using artificial neural network based on EEMD decomposition," *Environ. Res.*, vol. 139, pp. 46–54, May 2015.
- [175] C. Ma and Y. Li, "Improving forecasting accuracy of annual runoff time series using RBFN based on EEMD decomposition," *DEStech Trans. Eng. Technol. Res.*, Tech. Rep., 2016. doi: 10.12783/dtetr/emme2016/9816.
- [176] X. Zhang, Q. Zhang, G. Zhang, Z. Nie, and Z. Gui, "A hybrid model for annual runoff time series forecasting using Elman neural network with ensemble empirical mode decomposition," *Water*, vol. 10, no. 4, p. 416, 2018.
- [177] M. Niu, K. Gan, S. Sun, and F. Li, "Application of decomposition-ensemble learning paradigm with phase space reconstruction for day-ahead PM 2.5 concentration forecasting," *J. Environ. Manage.*, vol. 196, pp. 110–118, Jul. 2017.
- [178] J. Sun, C. Yan, and J. Wen, "Intelligent bearing fault diagnosis method combining compressed data acquisition and deep learning," *IEEE Trans. Instrum. Meas.*, vol. 67, no. 1, pp. 185–195, Jan. 2018.
- [179] J. Tian, C. Morillo, M. H. Azarian, and M. Pecht, "Motor bearing fault detection using spectral kurtosis-based feature extraction coupled with K-Nearest neighbor distance analysis," *IEEE Trans. Ind. Electron.*, vol. 63, no. 3, pp. 1793–1803, Mar. 2016.
- [180] D. Abboud, J. Antoni, S. Sieg-Zieba, and M. Eltabach, "Envelope analysis of rotating machine vibrations in variable speed conditions: A comprehensive treatment," *Mech. Syst. Signal Process.*, vol. 84, pp. 200–226, Feb. 2017.
- [181] Z. Wu and N. E. Huang, "Ensemble empirical mode decomposition: A noise-assisted data analysis method," *Adv. Adapt. Data Anal.*, vol. 1, no. 1, pp. 1–41, Jan. 2009.
- [182] N. A. Agana, E. Oleka, G. Awogbami, and A. Homaifar, "Short-term load forecasting based on a hybrid deep learning model," in *Proc. Southeast-Con*, Apr. 2018, pp. 1–6.
- [183] S. Maurya, V. Singh, and N. K. Verma, "Condition monitoring of machines using fused features from emd based local energy with DNN," *IEEE Sensors J.*, 2019.
- [184] F. Jia, Y. Lei, J. Lin, X. Zhou, and N. Lu, "Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data," *Mech. Syst. Signal Process.*, vols. 72–73, pp. 303–315, May 2016.
- [185] Z. Chen, S. Deng, X. Chen, C. Li, R.-V. Sanchez, and H. Qin, "Deep neural networks-based rolling bearing fault diagnosis," *Microelectron. Rel.*, vol. 75, pp. 327–333, Aug. 2017.
- [186] T. Junbo, L. Weining, A. Juneng, and W. Xueqian, "Fault diagnosis method study in roller bearing based on wavelet transform and stacked auto-encoder," in *Proc. 27th Chin. Control Decis. Conf. (CCDC)*, May 2015, pp. 4608–4613.
- [187] P. Wang *et al.*, "Virtualization and deep recognition for system fault classification," *J. Manuf. Syst.*, vol. 44, pp. 310–316, Jul. 2017.
- [188] X. Ding and Q. He, "Energy-fluctuated multiscale feature learning with deep ConvNet for intelligent spindle bearing fault diagnosis," *IEEE Trans. Instrum. Meas.*, vol. 66, no. 8, pp. 1926–1935, Aug. 2017.
- [189] J. Wang, J. Zhuang, L. Duan, and W. Cheng, "A multi-scale convolution neural network for featureless fault diagnosis," in *Proc. Int. Symp. Flexible Autom. (ISFA)*, Aug. 2016, pp. 65–70.
- [190] V. T. Tran, F. Althobiani, and A. Ball, "An approach to fault diagnosis of reciprocating compressor valves using Teager-Kaiser energy operator and deep belief networks," *Expert Syst. Appl.*, vol. 41, no. 9, pp. 4113–4122, Jul. 2014.

- [191] J. Wang, Y. Ma, L. Zhang, R. X. Gao, and D. Wu, "Deep learning for smart manufacturing: Methods and applications," *J. Manuf. Syst.*, vol. 48, pp. 144–156, Jul. 2018.
- [192] A. Harasimowicz, "Comparison of data preprocessing methods and the impact on auto-encoders performance in activity recognition domain," *ICT Young 2014*, p. 6, 2014.
- [193] D. Figo, P. C. Diniz, D. R. Ferreira, and J. M. P. Cardoso, "Preprocessing techniques for context recognition from accelerometer data," *Pers. Ubiquitous Comput.*, vol. 14, no. 7, pp. 645–662, Oct. 2010.
- [194] S. García, J. Luengo, and F. Herrera, *Data preprocessing data mining*. Springer, 2015.
- [195] T. Elsken, J. H. Metzen, and F. Hutter, "Neural architecture search: A survey," 2018, *arXiv:1808.05377*. [Online]. Available: <http://arxiv.org/abs/1808.05377>
- [196] S. Xie, H. Zheng, C. Liu, and L. Lin, "SNAS: Stochastic neural architecture search," 2018, *arXiv:1812.09926*. [Online]. Available: <http://arxiv.org/abs/1812.09926>
- [197] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," 2016, *arXiv:1611.01578*. [Online]. Available: <http://arxiv.org/abs/1611.01578>
- [198] C. Liu, B. Zoph, M. Neumann, J. Shlens, W. Hua, L.-J. Li, L. Fei-Fei, A. Yuille, J. Huang, and K. Murphy, "Progressive neural architecture search," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 19–34.
- [199] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
- [200] M. Lutter, C. Ritter, and J. Peters, "Deep Lagrangian networks: Using physics as model prior for deep learning," 2019, *arXiv:1907.04490*. [Online]. Available: <http://arxiv.org/abs/1907.04490>
- [201] P. Pillai, A. Kaushik, S. Bhavikatti, A. Roy, and V. Kumar, "A hybrid approach for fusing physics and data for failure prediction," *Int. J. Prognostics Health Manage.*, vol. 7, no. 025, pp. 1–12, 2016.
- [202] N. Bhutani, G. P. Rangaiah, and A. K. Ray, "First-principles, data-based, and hybrid modeling and optimization of an industrial hydrocracking unit," *Ind. Eng. Chem. Res.*, vol. 45, no. 23, pp. 7807–7816, Nov. 2006.
- [203] C.-C. Young, W.-C. Liu, and M.-C. Wu, "A physically based and machine learning hybrid approach for accurate rainfall-runoff modeling during extreme typhoon events," *Appl. Soft Comput.*, vol. 53, pp. 205–216, Apr. 2017.
- [204] J. Ling, A. Kurawski, and J. Templeton, "Reynolds averaged turbulence modelling using deep neural networks with embedded invariance," *J. Fluid Mech.*, vol. 807, pp. 155–166, Nov. 2016.
- [205] J.-S.-R. Jang, "ANFIS: Adaptive-network-based fuzzy inference system," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, no. 3, pp. 665–685, 1993.
- [206] J. Ling, R. Jones, and J. Templeton, "Machine learning strategies for systems with invariance properties," *J. Comput. Phys.*, vol. 318, pp. 22–35, Aug. 2016.
- [207] L. O. Chua and L. Yang, "Cellular neural networks: Theory," *IEEE Trans. Circuits Syst.*, vol. 35, no. 10, pp. 1257–1272, Oct. 1988.
- [208] G. Martius and C. H. Lampert, "Extrapolation and learning equations," 2016, *arXiv:1610.02995*. [Online]. Available: <http://arxiv.org/abs/1610.02995>
- [209] S. S. Sahoo, C. H. Lampert, and G. Martius, "Learning equations for extrapolation and control," 2018, *arXiv:1806.07259*. [Online]. Available: <http://arxiv.org/abs/1806.07259>
- [210] Z. Long, Y. Lu, and B. Dong, "PDE-net 2.0: Learning PDEs from data with a numeric-symbolic hybrid deep network," *J. Comput. Phys.*, vol. 399, Dec. 2019, Art. no. 108925.
- [211] K. Monakhova, J. Yurtsever, G. Kuo, N. Antipa, K. Yanny, and L. Waller, "Learned reconstructions for practical mask-based lensless imaging," *Opt. Express*, vol. 27, no. 20, p. 28075, Sep. 2019.
- [212] S. Boyd, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2010.
- [213] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Computer-Assist. Intervent.* Springer, 2015, pp. 234–241.
- [214] B. Yaman, S. A. H. Hosseini, S. Moeller, J. Ellermann, K. Uğurbil, and M. Akçakaya, "Self-supervised physics-based deep learning MRI reconstruction without fully-sampled data," 2019, *arXiv:1910.09116*. [Online]. Available: <http://arxiv.org/abs/1910.09116>
- [215] M. R. Kellman, E. Bostan, N. A. Repina, and L. Waller, "Physics-based learned design: Optimized coded-illumination for quantitative phase imaging," *IEEE Trans. Comput. Imag.*, vol. 5, no. 3, pp. 344–353, Sep. 2019.
- [216] C.-W. Chang and N. Dinh, "A study of physics-informed deep learning for system fluid dynamics closures," in *Proc. Amer. Nucl. Soc. Winter Meeting*, Anaheim, CA, USA, 2016, pp. 1785–1788.
- [217] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-Image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1125–1134.
- [218] A. Albert, E. Strano, J. Kaur, and M. Gonzalez, "Modeling global urbanization patterns with generative adversarial networks," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2018, pp. 1–4.
- [219] A. Albert, J. Kaur, E. Strano, and M. Gonzalez, "Spatial sensitivity analysis for urban land use prediction with physics-constrained conditional generative adversarial networks," 2019, *arXiv:1907.09543*. [Online]. Available: <http://arxiv.org/abs/1907.09543>
- [220] L. Sun, H. Gao, S. Pan, and J.-X. Wang, "Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data," 2019, *arXiv:1906.02382*. [Online]. Available: <http://arxiv.org/abs/1906.02382>
- [221] O. Andersson, F. Heintz, and P. Doherty, "Model-based reinforcement learning in continuous environments using real-time constrained optimization," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 2497–2503.
- [222] J. Scholz, M. Levihn, C. Isbell, and D. Wingate, "A physics-based model prior for object-oriented MDPs," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1089–1097.
- [223] D. Liu and Y. Wang, "Multi-fidelity physics-constrained neural network and its application in materials modeling," *J. Mech. Design*, vol. 141, no. 12, pp. 121403–1–121403–13, 2019.
- [224] N. Muralidhar, M. R. Islam, M. Marwah, A. Karpatne, and N. Ramakrishnan, "Incorporating prior domain knowledge into deep neural networks," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2018, pp. 36–45.
- [225] N. Muralidhar, J. Bu, Z. Cao, L. He, N. Ramakrishnan, D. Tafti, and A. Karpatne, "Physics-guided design and learning of neural networks for predicting drag force on particle suspensions in moving fluids," 2019, *arXiv:1911.04240*. [Online]. Available: <http://arxiv.org/abs/1911.04240>
- [226] C. J. Willmott and K. Matsuura, "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance," *Climate Res.*, vol. 30, no. 1, pp. 79–82, 2005.
- [227] F. Kaytez, M. C. Taplamacioglu, E. Cam, and F. Hardalac, "Forecasting electricity consumption: A comparison of regression analysis, neural networks and least squares support vector machines," *Int. J. Electr. Power Energy Syst.*, vol. 67, pp. 431–438, May 2015.
- [228] R. I. Levin, N. A. J. Lieven, and M. H. Lowenberg, "Measuring and improving neural network generalization for model updating," *J. Sound Vib.*, vol. 238, no. 3, pp. 401–424, Nov. 2000.
- [229] J. Benesty, J. Chen, Y. Huang, and I. Cohen, "Pearson correlation coefficient," in *Noise Reduction in Speech Processing*. Springer, 2009, pp. 1–4.
- [230] H. Xu, Z. Jiang, D. W. Apley, and W. Chen, "New metrics for validation of data-driven random process models in uncertainty quantification," *J. Verification, Validation Uncertainty Quantification*, vol. 1, no. 2, Jun. 2016, Art. no. 021002.
- [231] Y. Liu, W. Chen, P. Arendt, and H.-Z. Huang, "Toward a better understanding of model validation metrics," *J. Mech. Design*, vol. 133, no. 7, Jul. 2011, Art. no. 071005.
- [232] T. Toni, D. Welch, N. Strelkowa, A. Ipsen, and M. P. H. Stumpf, "Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems," *J. Roy. Soc. Interface*, vol. 6, no. 31, pp. 187–202, Feb. 2009.
- [233] N. M. Mangan, J. N. Kutz, S. L. Brunton, and J. L. Proctor, "Model selection for dynamical systems via sparse regression and information criteria," *Proc. Roy. Soc. A, Math., Phys. Eng. Sci.*, vol. 473, no. 2204, Aug. 2017, Art. no. 20170009.
- [234] K. P. Burnham and D. R. Anderson, "Multimodel inference: Understanding AIC and BIC in model selection," *Sociol. Methods Res.*, vol. 33, no. 2, pp. 261–304, Nov. 2004.
- [235] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proc. Nat. Acad. Sci. USA*, vol. 113, no. 15, pp. 3932–3937, Apr. 2016.
- [236] P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data," *J. Fluid Mech.*, vol. 656, pp. 5–28, Aug. 2010.
- [237] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz, "On dynamic mode decomposition: Theory and applications," 2013, *arXiv:1312.0041*. [Online]. Available: <http://arxiv.org/abs/1312.0041>
- [238] B. O. Koopman, "Hamiltonian systems and transformation in Hilbert space," *Proc. Nat. Acad. Sci. USA*, vol. 17, no. 5, pp. 315–318, May 1931.

- [239] I. Mezić, "Analysis of fluid flows via spectral properties of the Koopman operator," *Annu. Rev. Fluid Mech.*, vol. 45, no. 1, pp. 357–378, Jan. 2013.
- [240] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, "A data-driven approximation of the Koopman operator: Extending dynamic mode decomposition," *J. Nonlinear Sci.*, vol. 25, no. 6, pp. 1307–1346, Dec. 2015.
- [241] M. O. Williams, C. W. Rowley, and I. G. Kevrekidis, "A kernel-based approach to data-driven Koopman spectral analysis," 2014, *arXiv:1411.2260*. [Online]. Available: <http://arxiv.org/abs/1411.2260>
- [242] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Sparse identification of nonlinear dynamics with control (SINDy)," *IFAC-PapersOnLine*, vol. 49, no. 18, pp. 710–715, 2016.
- [243] J. L. Proctor, S. L. Brunton, and J. N. Kutz, "Dynamic mode decomposition with control," *SIAM J. Appl. Dyn. Syst.*, vol. 15, no. 1, pp. 142–161, 2016.
- [244] J. L. Proctor, S. L. Brunton, and J. N. Kutz, "Generalizing Koopman theory to allow for inputs and control," *SIAM J. Appl. Dyn. Syst.*, vol. 17, no. 1, pp. 909–930, Jan. 2018.
- [245] M. Schmidt and H. Lipson, "Distilling free-form natural laws from experimental data," *Science*, vol. 324, no. 5923, pp. 81–85, Apr. 2009.
- [246] Z. Long, Y. Lu, X. Ma, and B. Dong, "PDE-net: Learning PDEs from data," 2017, *arXiv:1710.09668*. [Online]. Available: <http://arxiv.org/abs/1710.09668>
- [247] W. Zaremba, K. Kurach, and R. Fergus, "Learning to discover efficient mathematical identities," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 1278–1286.
- [248] M. Raissi and G. E. Karniadakis, "Hidden physics models: Machine learning of nonlinear partial differential equations," *J. Comput. Phys.*, vol. 357, no. 1, pp. 125–141, Mar. 2018.
- [249] Z. Wang, S. Rosa, B. Yang, S. Wang, N. Trigoni, and A. Markham, "3D-PhysNet: Learning the intuitive physics of non-rigid object deformations," 2018, *arXiv:1805.00328*. [Online]. Available: <http://arxiv.org/abs/1805.00328>
- [250] G. Shi, X. Shi, M. O'Connell, R. Yu, K. Azizzadenesheli, A. Anandkumar, Y. Yue, and S.-J. Chung, "Neural lander: Stable drone landing control using learned dynamics," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 9784–9790.
- [251] M. Haghighatlar, G. Vishwakarma, M. A. F. Afzal, and J. Hachmann, "A physics-infused deep learning model for the prediction of refractive indices and its use for the large-scale screening of organic compound space," 2019.
- [252] M. Hilbert and P. López, "The World's technological capacity to store, communicate, and compute information," *Science*, vol. 332, no. 6025, pp. 60–65, Apr. 2011.
- [253] Y. Yu, H. Yao, and Y. Liu, "Physics-based learning for aircraft dynamics simulation," in *Proc. PHM Soc. Conf.*, vol. 10, 2018, pp. 1–9.
- [254] C. W. Anderson, "Learning to control an inverted pendulum using neural networks," *IEEE Control Syst. Mag.*, vol. 9, no. 3, pp. 31–37, Apr. 1989.
- [255] S. Adam, L. Busoniu, and R. Babuska, "Experience replay for real-time reinforcement learning control," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 2, pp. 201–212, Mar. 2012.
- [256] B. Settles, "Active learning literature survey," Dept. Comput. Sci., Univ. Wisconsin, Madison, WI, USA, Tech. Rep. 1648, 2009, pp. 1–44.
- [257] X. J. Zhu, "Semi-supervised learning literature survey," Dept. Comput. Sci., Univ. Wisconsin, Madison, WI, USA, Tech. Rep. 1530, 2005, pp. 1–38.
- [258] P. Donmez, G. Lebanon, and K. Balasubramanian, "Unsupervised supervised learning I: Estimating classification and regression errors without labels," *J. Mach. Learn. Res.*, vol. 11, pp. 1323–1351, Apr. 2010.
- [259] Y. Roh, G. Heo, and S. E. Whang, "A survey on data collection for machine learning: A big data-ai integration perspective," *IEEE Trans. Knowl. Data Eng.*, Aug. 2019. [Online]. Available: <https://arxiv.org/pdf/1811.03402.pdf>, doi: [10.1109/TKDE.2019.2946162](https://doi.org/10.1109/TKDE.2019.2946162).
- [260] S. Rao, S. Katoch, P. Turaga, A. Spanias, C. Tepedelenioglu, R. Ayyanar, H. Braun, J. Lee, U. Shanthamallu, M. Banavar, and D. Srinivasan, "A cyber-physical system approach for photovoltaic array monitoring and control," in *Proc. 8th Int. Conf. Inf., Intell., Syst. Appl. (IISA)*, Aug. 2017, pp. 1–6.
- [261] J. Wan, A. Canedo, and M. A. Al Faruque, "System level CPS design using non-Euclidean training method," Center Embedded Cyber-Phys. Syst., Univ. California Irvine, Irvine, CA, USA, Tech. Rep. TR#18-02, 2018.
- [262] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: Going beyond Euclidean data," *IEEE Signal Process. Mag.*, vol. 34, no. 4, pp. 18–42, Jul. 2017.
- [263] P. Li, J. Eickmeyer, and O. Niggemann, "Data driven condition monitoring of wind power plants using cluster analysis," in *Proc. Int. Conf. Cyber-Enabled Distrib. Comput. Knowl. Discovery*, Sep. 2015, pp. 131–136.
- [264] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," 2013, *arXiv:1312.6203*. [Online]. Available: <http://arxiv.org/abs/1312.6203>
- [265] M. Henaff, J. Bruna, and Y. LeCun, "Deep convolutional networks on graph-structured data," 2015, *arXiv:1506.05163*. [Online]. Available: <http://arxiv.org/abs/1506.05163>
- [266] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 3844–3852.
- [267] F. Scarselli, M. Gori, A. Chung Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2009.
- [268] J. Masci, D. Boscaini, M. M. Bronstein, and P. Vandergheynst, "Geodesic convolutional neural networks on Riemannian manifolds," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop (ICCVW)*, Dec. 2015, pp. 37–45.
- [269] V. Šnášel, J. Nowaková, F. Xhafa, and L. Barolli, "Geometrical and topological approaches to big data," *Future Gener. Comput. Syst.*, vol. 67, pp. 286–296, Feb. 2017.
- [270] J. M. Johnson and T. M. Khoshgftaar, "Survey on deep learning with class imbalance," *J. Big Data*, vol. 6, no. 1, p. 27, 2019.
- [271] N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study," *Intell. Data Anal.*, vol. 6, no. 5, pp. 429–449, Oct. 2002.
- [272] N. V. Chawla, N. Japkowicz, and A. Kotcz, "Editorial: Special issue on learning from imbalanced data sets," *ACM SIGKDD Explorations Newslett.*, vol. 6, no. 1, pp. 1–6, Jun. 2004.



RAHUL RAI received the B.Tech. degree from the National Institute of Foundry and Forge Technology (NIFFT), Ranchi, India, in 2000, the M.S. degree in manufacturing engineering from the Missouri University of Science and Technology (Missouri S&T), USA, in 2002, and the Ph.D. degree in mechanical engineering from The University of Texas at Austin, USA, in 2006. He is currently an Associate Professor of mechanical and aerospace engineering with the University at

Buffalo-SUNY.

His research is focused on developing computational tools for solving related problems in the manufacturing and system design domain. By combining engineering innovations with methods from machine learning, AI, statistics and optimization, and geometric reasoning, his research strives to solve important problems in the manufacturing and engineering design domain. His research has been supported by NSF, DARPA, ONR, NSWC, DMDII, HP, NYSEDA, and NYSPII. He has authored over 100 articles to date in peer-reviewed conferences and journals covering a wide array of problems. He was a recipient of the 2009 HP Design Innovation and the 2017 ASME IDETC/CIE Young Engineer Award. He also received the 2019 PHM Society Conference Best Paper Award.



CHANDAN K. SAHU received the B.Tech. degree from the National Institute of Technology, Rourkela, India, in 2014, and the M.S. degree in mechanical engineering from the State University of New York at Buffalo, USA, in 2020. He has also worked at Honda Motorcycle and Scooter, India, from 2014 to 2017. At Honda, he led the development automobiles with respect to their safety and emission aspects in close cooperation with SIAM, UNECE, EU, and JAMA. He is currently a Research Assistant with the Manufacturing and Design Laboratory (MADLab), University at Buffalo.

He is interested in machine learning in general, with a focus at their geometric attributes and mechanical applications. He has explored topological data analysis, tropical geometry, algebraic geometry, and so on. His current research work focuses on leveraging neural networks with physics models to enable better learning.

...