BOON OF HYBRID APPROACHES OVER THE BANE OF MODEL BASED AND MACHINE LEARNING APPROACHES IN MODELLING CYBER-PHYSICAL SYSTEMS

by

Chandan Kumar Sahu

February 2020

A dissertation submitted to the
Faculty of the Graduate School of
the University at Buffalo, The State University of New York
in partial fulfilment of the requirements for the
degree of

Master of Science

Department of Mechanical and Aerospace Engineering

ProQuest Number: 27735437

All rights reserved

INFORMATION TO ALL USERS
The quality of this reproduction is dependent on the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript
and there are missing pages, these will be noted. Also, if material had to be removed,
a note will indicate the deletion.

ProQuest 27735437

Published by ProQuest LLC (2020). Copyright of the Dissertation is held by the Author.

All Rights Reserved.
This work is protected against unauthorized copying under Title 17, United States Code
Microform Edition © ProQuest LLC.

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346

I grant The University at Buffalo the non-exclusive right to use this work for the University's own purposes and to make single copies of the work available to the public on a not-for-profit basis if copies are not otherwise available.

Chandan Kumar Sahu

Table of Contents

List of Figures **v**

Abstract **vi**

Chapter 1 Introduction **1**

Chapter 2 Modeling Cyber-Physical Systems **6**

2.1 Physics Equation Based CPS Modeling 8

2.2 State Machines Based CPS Modeling 10

2.3 Rule and agent-based CPS Modeling 11

2.4 Data-Driven CPS Modeling 13

Chapter 3 Challenges in Modeling Cyber-Physical Systems **15**

3.1 Discretization the continuum 16

3.2 Environmental Chaos and Limits of determinism of models . . . 18

3.3 Evolution from Newtonian systems to Mertonian systems 19

3.4 Advent of Big Data 20

Chapter 4 Synergy of Machine Learning and physics-based models **21**

Chapter 5 Hybrid Modeling: Infusing Physics into Data-driven Models **26**

5.1 Physics-based Preprocessing 29

5.2 Physics-Based Innovative Network Architectures 33

5.3 Physics-Based Constrained Regularization 37

5.4 Miscellaneous Architectures 39

Chapter 6 Metrics for Hybrid Models **42**

6.1 Universal Error Measures 42

6.1.1 Mean Square Error 43

6.1.2 Mean Absolute Error 44

6.2 Universal Metrics of a Model 44

6.2.1 Interpolability 45

6.2.2 Extrapolability 46

6.2.3 Sensitivity to the size of data 47

6.2.4 Robustness to Noisy Data 47

6.2.5 Model Complexity 48

Chapter 7 Directions for Future Work in Hybrid Modeling Domain **49**

7.1 Model Selection based on Hybrid Models 50

7.2 Benchmark Problems in Hybrid Domain 50

7.3 Regression Problems with low amount of data 51

7.4 Learning with Less Labels 52

7.5 Intelligent Collection of data for Hybrid Models 52

7.6 Geometric Machine Learning in Cyber-Physical Systems 54

7.7 Imbalanced Data and Data at Tail 54

Chapter 8 Conclusion **56**

Bibliography **58**

List of Figures

1.1 Schematic diagram of a Cyber-Physical System (CPS) 2

1.2 The spectrum of pure physics to pure data driven models for predicting the path of an oncoming quadrotor from a hexacopter. . . 4

2.1 A hypothetical state machine illustrating different states and transitions 11

4.1 Schematic diagram showing the current work in hybrid modeling 25

5.1 An integrated framework for showing all the three classes of hybrid architectures 28

5.2 Physics-based preprocessing architectures 30

5.3 Figure showing different physics-based preprocessing techniques used for hybrid modeling 33

5.4 Physics-based fusion architectures 34

5.5 Physics-based constrained regularization architectures 37

Abstract

Physics-based models (MB) and machine learning models (ML) have emerged as two paradigms for modeling any system. They represent the two opposite ends of the spectrum of system knowledge/data. Interpretability, tractability, and composability provided the thrust for MB. Genericity and accuracy drove ML. Interestingly, the drivers of physics models are the deterrents of machine learning models and vice versa. Even with their inherent boons and banes, they have evolved and practiced independently of each other for computing. Recently, several works attempted to merge MB and ML models for the complete exploitation of their combined potential. However, the research is scattered and unorganized. So, we make a meticulous attempt at organizing and standardizing the methods of combining ML and MB models. In addition to that, we prepared a five-faceted generic framework for the comprehensive evaluation of hybrid learning models (MB + ML). Finally, we conclude by shedding some light on the challenges of hybrid models, which we as a research community should focus for expediting the research and harnessing the full potential of hybrid learning models. All of these have been discussed with a clear focus on modeling cyber-physical systems (CPS) and overcoming the limitations of the existing models of CPS.

Chapter 1

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 mankind 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. ComputingScope will give us a vision of the whole and help us synthesize actionable information. The recent success of Deep Learning (DL) [1, 2, 3, 4, 5, 6, 7, 8] acts as a harbinger of the ComputingScope potential. 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.

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

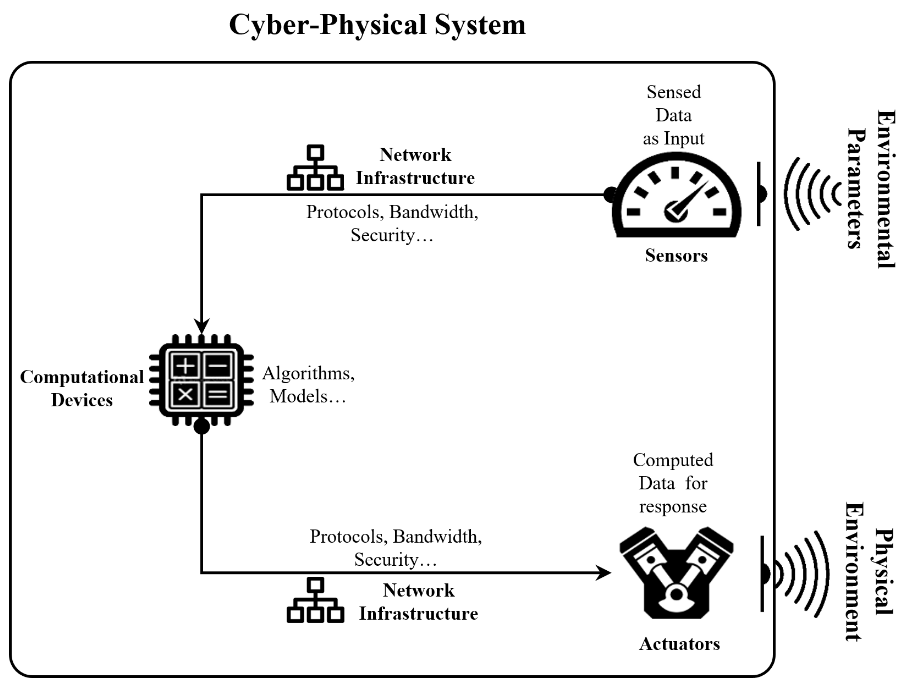

Figure 1.1: Schematic diagram of a Cyber-Physical System (CPS)

are complex systems capable of **C**ognition, **C**ommunication, **C**omputation and **C**ontrol (**4C**) [9, 10, 11, 12, 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 the stimuli, a computational intelligence platform for computing the response contingent upon the stimuli and a network that interconnects all of them. Fig. 1.1 illustrates the architecture and functioning of a CPS. It depicts 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. The development of intelligent *computational techniques (Computation CPS component)* that are capable of harnessing CPS sensor curated raw data to generate actionable intelligence (typically hidden in data) is the focus of this review.

A multitude of CPS applications including design, control, diagnosis, prognostics, and 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). Figure 1.2 shows the performance of pure MB, ML and hybrid models in a qualitative manner for predicting the path of an oncoming quadcopter from a hexacoptor. $(a), (c)\ and\ (b)$ represent the predicted path of pure physics, pure data-driven and hybrid models respectively. The corresponding models are shown in $(e), (g)\ and\ (f)$. Fig. (d) shows an instance of data which constitute the position $(\vec{p})$, orientation $(\vec{\theta})$ and velocity $(\vec{v})$ of the quadrotor.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

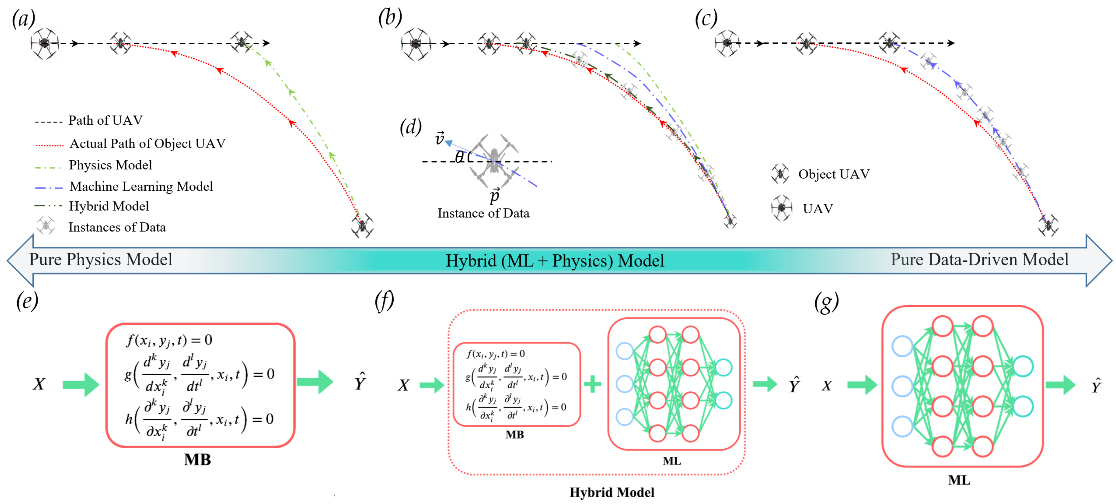

Figure 1.2: The spectrum of pure physics to pure data driven models for predicting the path of an oncoming quadrotor from a hexacopter.

enable predictive modeling of CPSs in the presence of imperfect models and sparse and noisy data.

Our review is complementary to existing review in 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 very 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 chapters with distinct themes. Chapter 2 discusses the existing techniques of modeling CPS. The hurdles faced by the existing CPS modeling techniques are elaborated in chapter 3. The necessity of combining ML and MB for eliminating those hurdles are analyzed in chapter 4. The methods for combining ML and MB models are discussed diligently in chapter 5. The metrics for wholesome evaluation of the hybrid models are deliberated in chapter 6. The new challenges which emanate from hybrid modeling are explored in chapter 7. And finally, chapter 8 concludes our work. Wherever possible, we have used a ball bouncing under gravity as a running example for maintaining consistency throughout the paper.

Chapter 2

Modeling Cyber-Physical Systems

A cyber-physical system (CPS) 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, 16, 17, 18, 19, 20]. The term 'Cyber-Physical Systems (CPS)' 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, 24, 25, 26], building control [27, 28, 29, 30], security [31, 32, 33], energy management [34], hazard response [35, 36, 37, 38], healthcare [16, 39, 40], society [41, 42], transportation [43, 44], environment [45] and manufacturing industries [46, 47, 48, 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 located in the following references [16, 31, 50, 51, 52, 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 etc. were 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 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.

Before we delve into different CPS modeling paradigms, it is important to delineate differences and commonality among terms such as Cyber-Physical Systems (CPS), wireless sensor networks (WSN), and Internet of Things (IoT). Although there is a significant overlap among CPS, WSN, and IoT, each of them serves a different purpose. CPS are capable of sensing the user or environment generated stimuli, processing the stimuli, creating a response using a computational framework, communicating it to the actuators and finally, the actuators respond to the creator of the stimuli. WSNs only sense the signal and emphasize on data collection. They can also extend their capabilities to data preprocessing, integration etc. IoT connect the WSN with internet and are capable of wholesome sensing, secure and reliable transmission, processing of the information. However, they are limited by actuation. To conclude, WSN is a subset of IoT, which in turn is a subset of CPS ($WSN \subset IoT \subset CPS$).

To harness the revolutionizing potential of the CPS, several attempts have been made at modeling CPS. Depending on the intended goal of the CPS system 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. The physics-based equations and state machines are used to model the continuous and discrete dynamics of a system. They are purely based on the physical dynamics of the system. They are robust but lack intelligence. Rule and agent-based systems leverage both data generated by the system and the knowledge of the system. They are relatively smarter. And hence more capable. The data-driven methods take complete advantage of data. These models are uber smart. For data-driven models, understanding of the system dynamics is not required even at the superficial level. Yet, they are the most intelligent. All these modeling paradigms are briefly discussed in the following sections.

2.1 Physics Equation Based CPS Modeling

The physical dynamics of all systems are continuous in nature. So does those of CPS. Physics-based equations effectively capture such continuous dynamics. However, the dynamics of every system is different. As a result, the physics-based equations which captures the dynamics will also be different for different systems. It is unlikely for two systems to share any common equation if they do not have any common dynamics. These physics-based models usually take the form of equations. The equations can be simple dependence equations or ordinary differential equations or partial differential equations or a combination of them. For a time(t) dependent system of n independent variables $x_1, x_2, ...x_n$ and a single dependent variable y, a simple dependence equation can be of the form

$$f(t, x_1, x_2, ...x_n) = y \tag{2.1}$$

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

$$f(t, x, y, \frac{dy}{dt}, \frac{dy}{dx}, ...\frac{d^n y}{dt^n}, \frac{d^n y}{dx^n}) = 0 \tag{2.2}$$

In the case of complex systems, it is difficult to identify the individual impact of a parameter accurately. In such cases, the dynamics take the form of partial differential equations. Then the equations will be

$$f(x, y, \frac{\partial y}{\partial t}, \frac{\partial y}{\partial x}, ...\frac{\partial^n y}{\partial t^n}, \frac{\partial^n y}{\partial x^n}) = 0 \tag{2.3}$$

The equations (2.1) - (2.3) consider only one target variable. They should be appropriately modified for systems with multiple target variables. With a deeper investigation of the system, detailed dynamics can be discovered. That leads to more detailed equations.

Irrespective of whether the equation includes simple or ordinary or partial derivatives, all three kinds can only be formulated for continuous functions. If the function is not continuous in its domain, the derivative can't be computed. So, such equation-based methods (2.1)-(2.3) are limited to continuous dynamics.

As every component of a system can have a different set of equations, the component is represented as an *actor* with equation-based models explicit to it. Such *actors* are congregated as *actor models* to model a system's continuous dynamics [9]. The best feature of the physics-based equations is that causative relationships can be established among the system variables and parameters. It offers better control of the system. However, the behavior of a system is not necessarily smooth all the time e.g., collisions of mechanical components. In such cases, the behavior can be modeled using modal models that are used for modeling discrete, instantaneous events.

2.2 State Machines Based CPS Modeling

State machines are used for modeling discrete dynamics. 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 ordinarily abrupt. Discrete events are instantaneous in nature. So, the derivative is zero before and after the event but tends to hit infinity at the instant. That's why they can't be modeled using equations. 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]. 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 [55]. If the output is produced when the machine is in a particular state, it is a Moore machine [56]. They are convertible into each other. A simple hypothetical finite state machine model is shown in fig. 2.1. It also shows different kinds of state transitions.

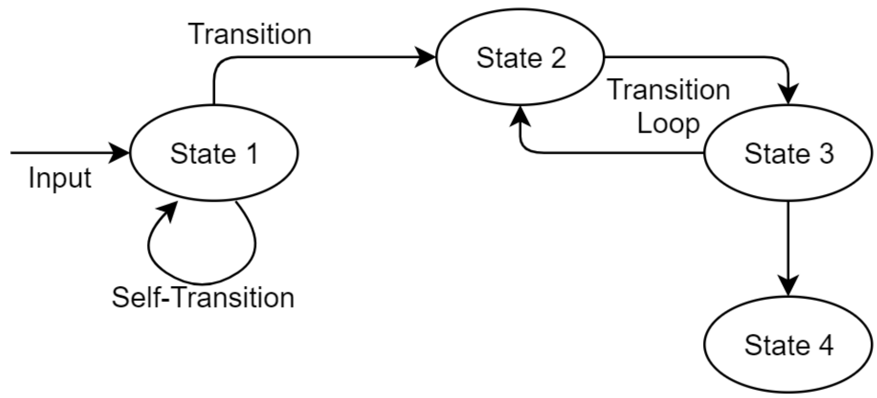

Figure 2.1: A hypothetical state machine illustrating different states and transitions

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.

2.3 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 inter-dependencies 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) [57, 58, 59] and *agent-based systems* (ABM) [60, 61]. 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 [60]. MAS is supported by a set of sensors for making them aware of the situation. The sensors sense the environment in an event or time-triggered manner and create data. The data is acted upon by smart *agent(s)* using decision-making algorithms. The agents make decisions based on a semantically defined set of *rules*. As they use data and semantic rules, MAS are capable of extraction of insightful information from the data. It makes the agents smart and intelligent. Temporal continuity, social skills and autonomy are the main attributes of an agent [62, 63]. 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 etc.). 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. Ontology plays a vital role in modeling agent-specific and inter-agent dynamics in MAS. The ontology and semantics are usually modeled using an object-oriented program. It helps in reducing redundant information.

Unlike the physics-based or state machine-based models, which are limited by the need for adaptability and inter-operability [63] 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 [64] like those in physics-based or state machine based models.

2.4 Data-Driven CPS Modeling

The data-driven models completely rely on data. They obviate the knowledge of system dynamics. As they disregard the system dynamics, handling heterogeneity and complexity of the CPS becomes a cakewalk. The data-driven models must be able to model the inherent characteristics of the CPS, i.e., continuous and discrete dynamics of the CPS. This can be rephrased as the data-driven models must be able to accomplish regression and classification tasks. Thus they will be able to capture the hybrid nature of the CPS.

Attempts at data-driven modeling of CPS for capturing the hybrid dynamics tried to exploit concepts from bayesian learning [65], algebraic geometry [66], bounded error [67], compressive sensing [68], inference of state transition logics [69], regression approach underpinned by recursive clustering and linear multi-class classification [70], symbolic regression [71], hybrid stable spline algorithms [72] etc. An excellent review on earlier methods of identifying hybrid systems can be found in [73]. The earlier methods [65, 66, 67, 74] mostly focused on simple piecewise affine systems which use linear transition rules[69]. Recent methods try to capture the non-linear dynamics from relatively complex systems [68, 69, 70, 71, 72]. Recently, deep neural networks with non-linear activation functions received widespread recognition for their ability to learn complex non-linear functions [75, 76, 77, 78]. They have also started to find use in modeling CPS [79, 80]. Currently, there is a consensus that only models powered by artificial intelligence can achieve acceptable levels of autonomy by coping up with uncertainties of the real world [81]. The deep neural networks, 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, which is an important deliverable of CPS. Nonetheless, the deep learning techniques are plagued by their own limitations, which are briefly discussed in chapter 4. Hence, the current work focuses on combining conventional physics knowledge with neural networks to overcome those limitations.

The primary reasons for 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.

Chapter 3

Challenges in Modeling Cyber-Physical Systems

Model-based designs have been dominating the sphere of CPS models until the entry of data-driven models into it. 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 were the main drivers behind their success. Interpretability prevented failures. Composability facilitated the construction of complex CPS. Together they made the physics-based models of CPS ubiquitous and indispensable. But, the model-based systems are plagued by a diverse set of challenges. They include solver dependency of the model, Zeno behavior, non-determinacy, heterogeneity of the system, maintaining the consistency among the components of the model, distributed behavior, misconnected model components [82] etc. Different numerical solvers used in a CPS model may use different numerical precision. Again, the solvers dynamically adjust the precision of time. It can affect the behavior of the CPS model [82]. The time precision of the model must be able to capture every time instant when an event occurs. As simple components evolve into complex systems, understanding and modeling the interaction among the components becomes challenging. Ensuring the 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 [82]. The heterogeneity arising due to different components of a CPS, which necessitates the blending of diverse domain specific-knowledge must also be taken care of. However, our goal is not to discuss the methods to address all the challenges mentioned above. Rather, we briefly delve into only those specific challenges of CPS models, which can be eradicated or minimized by combining model-based and data-driven methods.

3.1 Discretization the continuum

In cyber-physical systems, cyber systems like microprocessors and digital communication networks portray discrete dynamics. Whereas, the physical elements of the CPS like electro-mechanical, humane, 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 the 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 systems using some means like numerical approximation and numerical integration. 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 system and that's the usual case.

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 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. E.g., 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 of 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 [83]. 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 as *Zeno point*. Although challenging [83], there are some prior work that attempted to determine if the system is going to portray Zeno behavior and the Zeno point of such systems [84], but there is a dearth of work on modeling Zeno behavior.

3.2 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 limit of determinism of the models [83]. Deterministic systems are predictable i.e. the output can be predicted accurately if the inputs, system parameters and the initial conditions are known precisely. However, most nonlinear feedback systems encounter chaos. Even real-time systems that seem robust, non-chaotic, like the standard scheduling algorithms, can exhibit chaotic behavior [85]. Almost all modeling frameworks exhibit chaos. Chaotic behavior is so fundamental that it weakens determinism. Determinism is a property of the CPS model which models a CPS system. It's an invaluable property as it facilitates definitive analysis. And henceforth, it enables very complex and reliable designs. 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. 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.

3.3 Evolution from Newtonian systems to Mertonian systems

With the advent of data-driven modeling, proliferation of Merton's systems has drastically increased than those of Newton's systems. It is only because of the close interaction between the actual and the artificial life. This is possible because of the support of big data, cloud computing, internet of things, neural networks, reinforcement learning etc. [86]. This evolution can be broadly restated as traditional systems with small data and big Newtonian laws have been 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. With human interaction and intervention, cyber-physical systems have become cyber-physical social systems. 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 [87, 88]. Such systems are governed by free will. Hence, they cannot be controlled but can only be influenced to achieve the desirable objective in a probabilistic manner. 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 case of Merton's systems, causality is a luxury which is almost unachievable with the available resources for complexity, diversity and uncertainty. 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.

3.4 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. Instead, they need a deeper understanding of the causal relationship of the system parameters and variables. So, they are effective at modeling any systems of simple to moderate complexity. More importantly, they are quite effective at modeling systems with small and 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, large size of data helps. It is only possible because of pervasive sensing, rapid computational capabilities, proliferation of the internet and its users etc. That surge in harnessing data 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.

Chapter 4

Synergy of Machine Learning and physics-based models

Both the ML models and physics-based models model a system. As stated previously, the MB do not need data to model the system. They need the complete understanding of the system dynamics. The system has to be simulated for understanding the system dynamics. Data is a by-product of those simulations. So, the data is a reflection of the system. MB need data for calibrating their effectiveness. On the other hand, ML models need data for modeling the system. Data without any consideration of the system dynamics empowers the ML methods to be applicable for all kinds of systems. Precisely, ML models capture the reflection of the system. However, data can never be complete because of the discrete nature of the input parameters and the output variables of the system. Irrespective of whether data acquisition is expensive or not, data can never fill the state space completely. So, the chasm between the system and the data generated by it is ingrained. This is evident from the success of CNNs with the development of ImageNet [89, 90] database, which were prone to failure with older databases [91, 92]. 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 paying any attention to 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's a twinborn problem at the expense of accuracy and generic nature of the ML models.

MB have the potential to actually 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 also be monetarily, computationally and **time** expensive. So, some dynamics are often ignored. 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. E.g. The impact of wind drag, energy lost due to non-elastic nature of collisions, elasticity of the material of the ball, buoyancy due to the medium etc. are often ignored in case of the bouncing ball. If simulations are carried out delicately and observed meticulously for capturing the dynamics accurately, then the MB can become complex. That complexity compromises tractability. Tractability is of paramount importance in control applications. Even sometimes, the impact of some dynamics are consciously ignored to make the model amenable. That makes accurate MB a double-edged sword.

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 which is going to model the system. Data is sparse. But, ML models discover a continuous function to fit the same sparse data. Thereby filling 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 trajectory. 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 model can model it nor the MB can be validated against it. So, it is essential for the data to capture all the events. The same goes for the 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.

Use of ML models eliminate the impact of heterogeneity and complexity of the CPS. Complex CPS can be modeled with ease. The interaction among the components can also be easily modeled using ML models if the data can be captured. If not, the dynamics of the system can be modeled directly by ignoring the impact of the dynamics at the component level. As ML models learn the features by themselves, the impact of asynchrony in time, delay and loss in the network need not be explicitly modeled. With the same data, ML models can be trained for both regression and classification tasks. Thus, they can be used to capture hybrid dynamics. As the ML models develop a continuous state space from discrete sparse data, environmental chaos are automatically captured in the model. All these advantages are driven by data. So, for the data-driven ML model to be really effective, the data should capture all the events. The time precision of the sensors should permit capturing all the events. Unless simulated, the events can't be captured. So, data-driven models will need infinite simulations to capture the complete behavior of the system. It is an impossible task. Even if it is possible, as data is a reflection of the system, the reflection will be complete but could be spurious. Moreover, data acquisition becomes impossibly difficult when the system exhibits Zeno behavior. For the case of the bouncing ball, the driving force is the initial height of the ball prior to the first free fall. Data about the bouncing height can't be acquired after Zeno point. That prohibits the ML models to model the ball's behavior beyond the Zeno point. 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, viscosity of the medium, non-elastic collisions, deformable nature of the material of the ball etc. contribute to the inhibiting forces. The buoyancy of the medium is a driving force. These aspects of the system can only be brought into the model only by using the MB. Hence, it is prudent to complement the ML with MB. It will make a complete, cogent and accurate model. So, by combining them, we can actually recover a complete model of the system. Hybrid models harness the best of both the worlds. We feed the knowledge we have into the model. The ML model discovers the knowledge which we don't know [93]. Previously, [94] outlined the advantages of the combination of model-based and data-driven models and attempted at formalizing the steps in hybrid modeling. We aim to standardize the combination and evaluation processes. We also present a succinct overview of the challenges which should be addressed by the community for accelerating the research in CPS. These are summarized in fig 4.1.

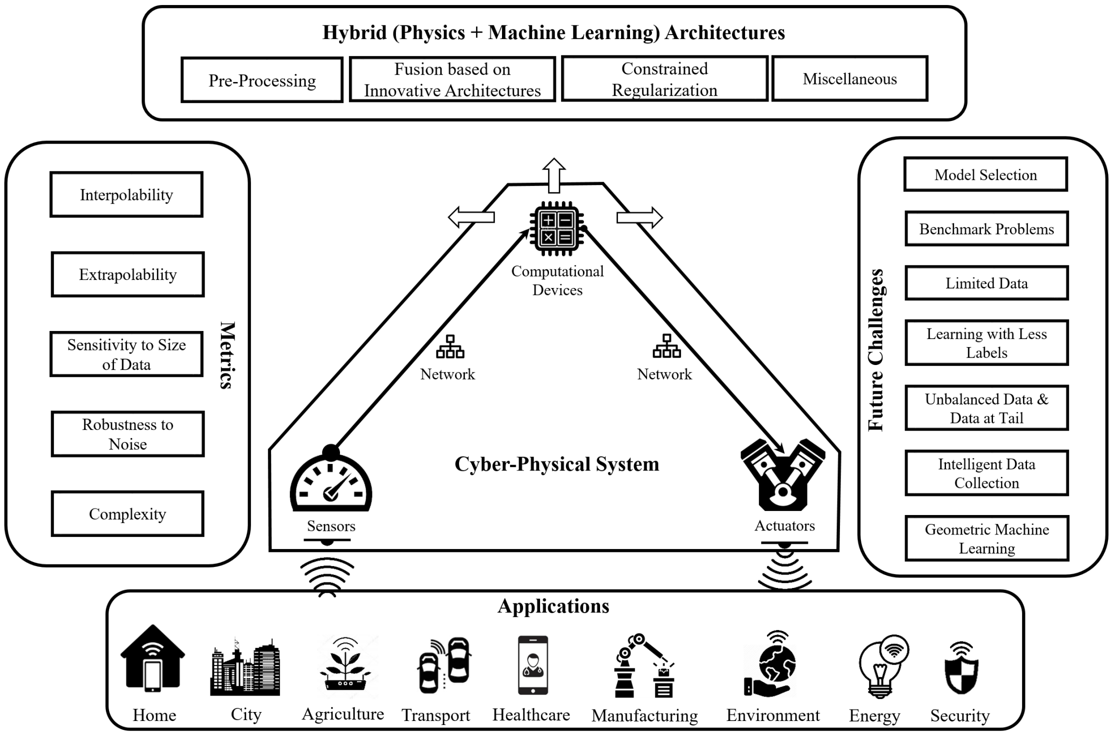

Figure 4.1: Schematic diagram showing the current work in hybrid modeling

Chapter 5

Hybrid Modeling: Infusing Physics into Data-driven Models

Consider a CPS predictive modeling task such as state assessment of a diesel engine or a UAV. Given input state variables $\mathbf{X}_{MB}$, model parameters $\mathbf{P}_{MB}$, and target variable Y, the physics-based models, $f_{MB} : [\mathbf{X}_{MB}, \mathbf{P}_{MB}] \rightarrow Y$ maps input state variables and parameter values to the target variable Y. Predictive modeling in purely model-based approaches are 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. The data-driven machine learning approach, however, is focused on training a model

$f_{ML} : \mathbf{X}_{ML} \rightarrow Y$ over a set of training data to produce estimates of outputs $\hat{Y}$ given inputs $\mathbf{X}_{ML}$. Both these approaches have their limitations.

In hybrid modeling approaches, one creates a hybrid combination of MB and ML approaches that combine the power of both (Figure 5.1). One way to combine them is to use the output of f_{MB} as an input to f_{ML} or $f_{Hybrid} : [\mathbf{X}_{MB}, \mathbf{P}_{MB}, \mathbf{P}_{ML}] \rightarrow \mathbf{Y}$, where $f_{Hybrid} = f_{ML} \circ f_{MB}$. The machine learning component will thus encapsulate the remaining unmodeled complexity of a system in a lumped form. In this approach, 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 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 $\mathbf{P}_{MB}$ (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', 'physical priors', 'physics-guided', 'residual physics', 'physics-based', 'physics fusion', 'physics-infused', 'theory-guided', 'hybrid', 'physics-informed', 'physics constrained', 'theory guided' etc. Most of the work are in computer vision, fluid and thermal applications. But, the work on combining MB and ML models are almost non-existent in cyber-physical applications. Hence, we expand the range of our search beyond the purview of cyber-physical applications to all available hybrid models. With our experience we can profess that the hybrid models share a lot of commonalities irrespective of the name using which they are referred. Depending on the nature of 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 (Figure 5.1) as follows 1) approaches that use P_{MB} as preprocessing or inputs to machine learning model (such as Deep Learning (DL)), 2) innovative network architectures that imbibe physics of the problem at hand in P_{ML}, and 3) loss functions that enforces physics infusion in the learned DL model.

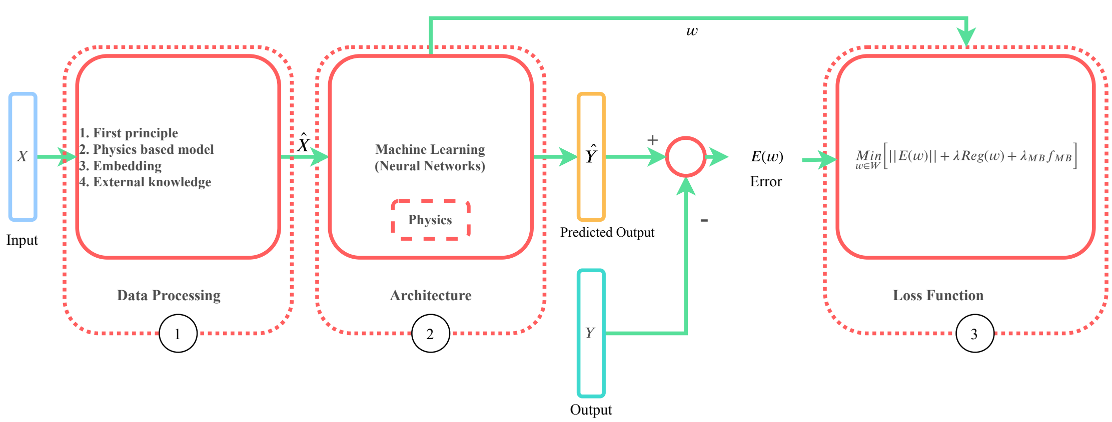

Figure 5.1: An integrated framework for showing all the three classes of hybrid architectures

The three major classes are motivated by the control levers of a deep neural network (DNN) for learning better functions to obtain accurate mappings. The three levers are (1) better quality of data which translates to better preprocessing techniques, (2) better feature extraction processes which translates to better NN architectures and (3) better loss functions which has been consistent so far. That covers all the control levers of a DNN for creating effective learning algorithms. We lump other existing work in 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 (Neural Networks). However, similar ideas could also be used in conjunction with other ML approaches such as Support Vector Machines (SVM) [95, 96, 97] and Random Forest (RF) [98, 99] etc.

5.1 Physics-based Preprocessing

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 succeeding processes. Noise removal, data transformation, data reduction, instance selection, dimension reduction etc. constitute the preprocessing techniques [100]. Unlike the feature extraction techniques, which could be very specific, the preprocessing techniques can be used in a wide variety of applications [101]. Still, there are application specific preprocessing techniques which are not the focus of our work. Preprocessing is essential for ML models. Normalization is the simplest form of preprocessing. Band-pass filtering, downsampling [102], channel selection, time window setting [103], data cleaning, encoding [104] are a few more simple preprocessing techniques to quote. To generalize, in this 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 processed data $\hat{x}$ and the actual outputs y and predicts the actual output as $\hat{y}$ with an error ϵ_i defined as $\epsilon_i = y - \hat{y}$. This class of hybrid models is pictorially represented in fig. 5.2

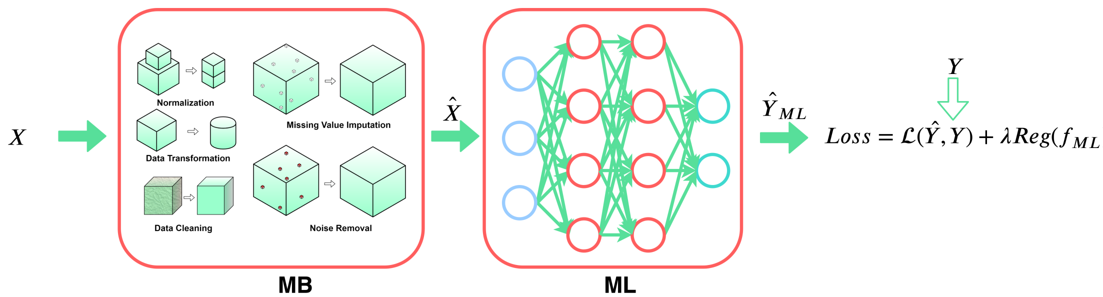

Figure 5.2: Physics-based preprocessing architectures

As the DNNs are capable of learning from pristine data, they have almost eliminated the need of preprocessing of data. 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 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 [105]. For non-stationary signals, time-frequency representations are used. The time domain and frequency domain representations are one-dimensional (1D) representations. Wavelet transform, empirical mode decomposition, short-time Fourier transform can be used to convert the 1D representations into 2D time-frequency domain representations [106]. Correspondingly, 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 [107], kurtosis estimation etc. fall under time domain analysis. Fast Fourier transform, bispectrum analysis etc. are a few examples of frequency domain analysis. Short-time Fourier transform (STFT), wavelet transform [108, 109], Hibert-Huang transform [109], sparse decomposition [110], wavelet packet transform etc. comes under the umbrella of time-frequency based preprocessing techniques [111].

Hybrid architectures which use physics-based preprocessing have found applications in manufacturing industry [112, 113, 114, 115], healthcare industry [116, 117, 118], energy industry[119], for processing hydrological data [120, 121, 122], meteorological data[123] among several others. The frequency spectrum of raw data was fed to a DNN based stacked auto-encoder [124]. 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 [125]. Compressed sensing was used for extracting low dimensional features from raw temporal data before passing it into a stacked denoising autoencoder DNN [126]. nonlinear soft threshold and digital wavelet frame were used to denoise the data before feeding it into a stacked auto-encoder [127]. Wavelet analysis was used for converting raw data into 2D time-frequency images, which were subsequently fed into a deep CNN [128]. Data transformed as wavelet packet energy image was supplied to a CNN [129]. The signal was decomposed using Morlet wavelet to get a multi-scale spectrogram image [130]. It was passed into a CNN for further processing. Deep Boltzmann machines usually accept data preprocessed by wavelet transform or Teager-kaiser energy[131] operator [132]. Other like spectral kurtosis [133] and envelope analysis [134] were used for extracting the frequency band with the most useful information. The output was further processed by using fast Fourier transform before feeding it to a CNN [112, 113]. Ensemble empirical mode decomposition (EEMD)[135] was used for decomposing the input data into a set of intrinsic mode functions (IMF) and a residual series [120]. 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 and Elman NN [122] , to an auto-encoder [116], into a LSTM [119] and a deep belief network[136]. [137] used EMD for extracting low level features and a DNN for extracting high level features. A computationally inexpensive delayed error normalized least mean square (DENLMS) [117] filter was used for preprocessing the data [117]. Additionally, discrete wavelet transform was used for extracting higher level features like heart rate variability (HRT) before passing the data to the ML model [117]. 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 an RNN [115]. The physics-based preprocessing techniques are pictorially represented in fig. 5.3 [138] studied the impact of a few preprocessing techniques like PCA, scaling, normalization etc. on the performance of autoencoders. [139] offered a detailed review on several preprocessing techniques like envelope metrics, statistical metrics, frequency and discrete domain metrics for feature extraction. For a comprehensive review of preprocessing techniques [140] can be referred.

Although the preprocessing techniques encapsulate a wide variety of goals like noise removal, data transformation, feature extraction etc., most of them are

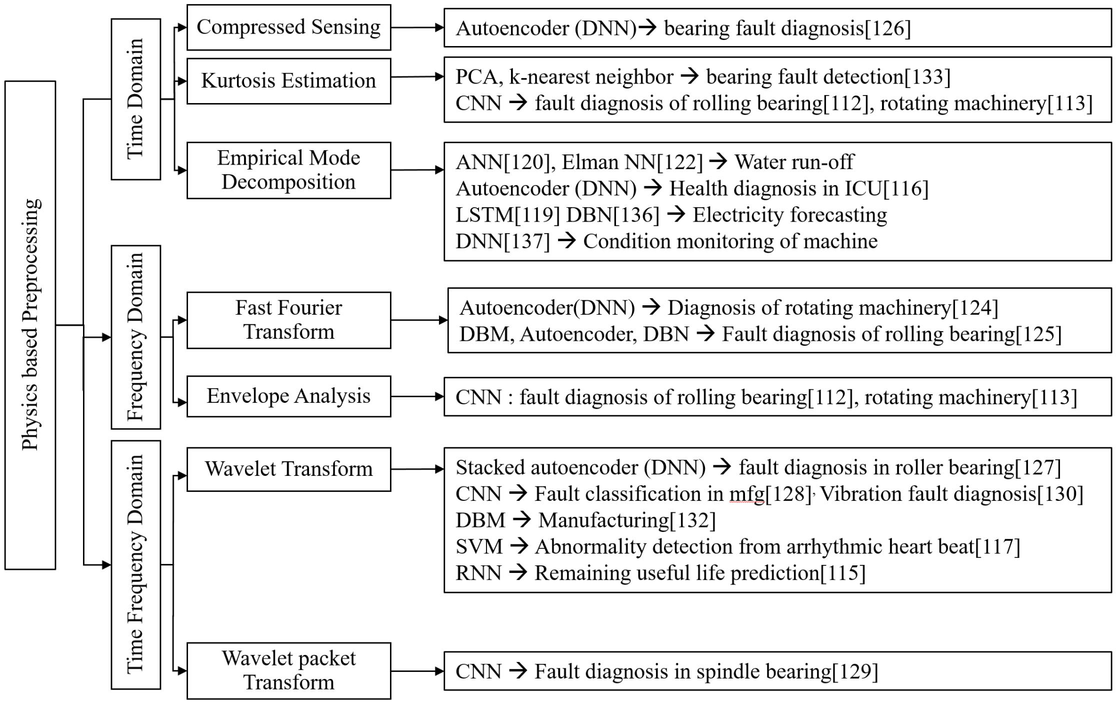

Figure 5.3: Figure showing different physics-based preprocessing techniques used for hybrid modeling

shallow features. They do not add any system-specific information to the ML model. To incorporate features of superior quality that offer better insight into the dynamics of the CPS, innovative architectures shall be devised.

5.2 Physics-Based Innovative Network Architectures

This kind of hybrid architectures try to explore for the best NN architecture, which can utilize the MB completely. A plethora of such architectures have 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 [141, 142, 143, 144]. Broadly, the architectures can be one or a combination of deep neural networks, convolutional neural networks, recurrent neural networks and other neural networks with physics. The innovative architectures can only ingest the output of the physics model into a particular node of a specific layer. Or the output of the MB can be embedded into the NN. In addition to the output of the MB, they can also take raw input data as another input. The fusion architectures are shown in fig. 5.4.

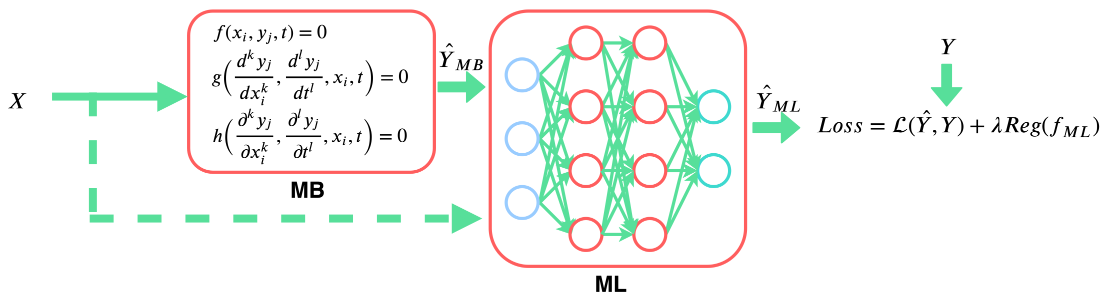

Figure 5.4: Physics-based fusion architectures

Such physics-based architectures have found use in environmental data analysis [145], dynamical systems [146], biological systems [146] and image processing [147]. [145] used the output of the physics model as an additional input feature for a multilayered perceptron model for modeling the temperature of a lake. [146] fed the output of the MB in the last layer of an LSTM network along with the unprocessed inputs for modeling an inverted pendulum and growth of cancer cells. The physics-based motion model of aerial vehicles was sandwiched between two RNN modules for multistep prediction of their nonlinear dynamics [148]. 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 [149] were formulated for creating a physical simulator offering universal uncertainty estimates. In TossingBot [150], the output of the physics-based controller was fed to FCN ResNet-7 [151] for robotic applications. A deep lagrangian network (DeLAN) [152] 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 [153]. A combination of series and parallel connection was used for predicting the failure of a turbine blade [153]. The MB was fused with an ANN and SVR [154] for modeling rainfall-runoff. [147] attempted at recovering 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 [155] 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.

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. Such deep embedded hybrid models have used in fluid flows [156, 157]. 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 the physics in the hybrid architecture. They developed the architecture for both continuous and discrete time models. This architecture was used for both solving [156] and discovering [157] nonlinear PDEs. Invariance or symmetry attributes were embedded into a NN [158] 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 to an ML model. An MB transformed as a NN coalesces better with a NN. A cellular NN [159] was used to solve the PDE in Hybrid-Net[160] before attaching it to a ConvLSTM network for fluid and thermal applications. A symbolic multilayered neural network inspired from equation learner [161, 162] was used for blending prior knowledge in [163] for learning accurate PDEs from data. In [164], the *LEarned Alternating Direction Method of Multipliers* (Le-ADMM) [165] method for solving inverse problems (MB) is unrolled into a NN. It was appended to another pure NN or U-Net [166] for reconstructing mask-based lenseless images into traditional images from lensed cameras. Such physics-based unrolled network has also been used for image reconstruction [167] and optimizing coded-illumination [168] 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 [169] model has been developed for predicting the trajectory of a projectile and for estimating the velocities of rigid objects after a collision.

Like the pure NN which learns the features by themselves, these architectures have complete freedom to learn the physical dynamics by themselves. It could be beneficial or detrimental. It is beneficial because it can discover unknown laws of physics from the data, because that's an innate ability of NN. However, that's not assured. Although they can discover new laws, no expertise is out there which can make them abide to the known laws of physics. So, it can be detrimental. That shoves us into the next class of hybrid models in which physics is hardwired into the NN.

5.3 Physics-Based Constrained Regularization

In this class of architectures, 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. The physics-based preprocessing and fusion architectures need the MB to be evaluated or require the the PDEs to be solved. But, regularization based approaches directly engulf the equation as a constraint. Thereby altering the computational cost. It is shown in fig. 5.5.

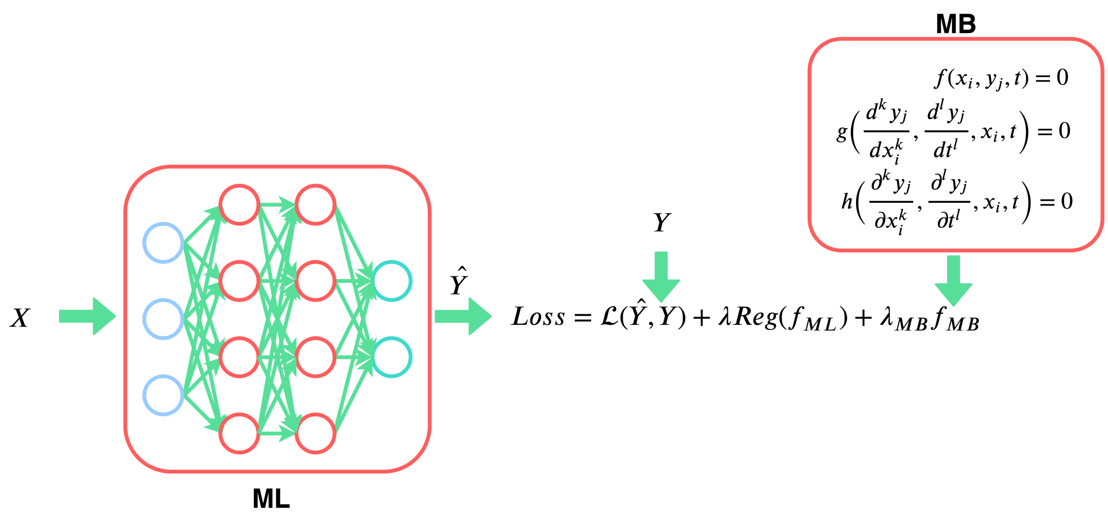

Figure 5.5: Physics-based constrained regularization architectures

Such hybrid models have been used for analysis of environmental data [145], machine vision applications [170] and fluid dynamics [171]. Temperature-density and density-depth relations were used as constraints for modeling the temperature of a lake at different depths [145]. In [170], a weighted constraint function was used to penalize data that are inconsistent with prior knowledge in case of the video of a projectile. The loss function of the *pix2pix* architecture in a conditional GAN [172, 173] was penalized by using a constraint enforcing module for spatial sensitivity analysis for predicting urban land use [174]. The loss of a fully connected NN was made to include boundary conditions and residuals of the governing equations for surrogate modeling in hemodynamics [175]. Reinforcement learning was also used for learning non-parametric models under constrained state spaces in continuous environments [176]. It was used for dynamic applications like quadcopter navigation and cart-pole problem. Physics has also been used as a prior in reinforcement learning [177]. As low fidelity MB's capture the general trend and the high fidelity MB's additionally capture local intricate details, a multi-fidelity physics constrained NN was proposed for incorporating linear and nonlinear PDEs [178]. Approximation and monotonicity constraints were also incorporated into the loss function of a NN [179] 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 [180]. An architecture with both fusion and constrained regularization was used in [181] for multiphase flows. It also included an additional loss term for the difference between the MB output and the actual output. It performs well under data scarcity. [170] demonstrated supervising a NN merely using the physics-based constraint using data without any labels.

5.4 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 guarantee learning of 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 ($g_{earth} = 9.8\ ms^{-2}$) with the same ball bouncing in a viscous medium on moon ($g_{moon} = 1.625\ ms^{-2}$). The energy lost while bouncing on earth is purely due to the gravity of earth. On moon, it will split into the energy lost due to moon's gravity and the viscosity of the medium. Consider that the viscosity of the medium is controlled on 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 will learn the same mapping in both the cases. It is because the ML models disregard the underlying physics and the incompleteness of data because of not recording the influence of other influencing factors. In moon's case, the ML model ignored the effect of viscosity which accounted for the difference between the accelerations due to gravity of moon and 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. 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) [182] 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) [183, 184] built on proper orthogonal decomposition (POD) can be used to extract spatio-temporal correlations. It tries to infer a linear operator at least for small time periods that approximates the spatial data, which can have millions degrees of freedom. Koopman operator [185], an infinite dimensional linear operator, was used for extending DMD for nonlinear dynamics [184, 186]. The inefficiency of linear operators for modeling the nonlinear dynamics led to Extended Dynamic Mode Decomposition (eDMD) [187, 188]. They have also been expanded for control applications in [189, 190, 191]. In addition to these, the equation learner [161, 162] which uses interpretable activation functions (sine, cosine, identity etc.) in a NN, genetic programming based [192], PDE-Nets [163, 193] based on the correlation between convolutional filters and differential operators are phenomenal. Two deep neural networks were used for discovering nonlinear dynamics in terms of nonlinear PDEs for spatio-temporal data [194]. 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 [195], 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 [196], the dynamics of a UAV was learned using NN and the MB was used for its control.

Chapter 6

Metrics for Hybrid Models

Error metrics are the 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 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. We have split this chapter into two sections. The first section is fundamental to the error metrics proposed in the succeeding section. Section 6.1 critically appraises the error measures. We have compared the mean square error (MSE) and mean absolute error and answered why we selected the MSE as our measure. Then, we exclusively use MSE in section 6.2 for gauging the performance of the models.

6.1 Universal Error Measures

The current sections discuss the error measures, which are fundamental for 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 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 $(= \sum_{i=1}^{i=n} \epsilon_i)$ is the sum of the individual errors ϵ_i, defined as $\epsilon_i = y_i - f_{Hybrid}(x_i)$. There are several ways to define the total error. Mean absolute error, mean absolute percentage error, Pearson correlation coefficient, mean square error etc. are a few to name. However, the mean square error (MSE) and the mean absolute error (MAE) are the most prominent ones, which are defined below:

6.1.1 Mean Square Error

The mean square error (RMSE) 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

$$MSE = \frac{1}{n} \sum_{i=1}^{i=n} [y_i - f_{Hybrid}(x_i)]^2 \tag{6.1}$$

It is closely related to the quadratic loss function and the p-norm. 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 the individual errors due to bias and variance. This is why the mean squared error (MSE) has taken the canonical form.

It heavily penalizes the outliers. All p-norms with $p > 1$ showcase similar behavior. That is good in 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.

6.1.2 Mean Absolute Error

The mean absolute error (MAE) is the main contender against the root 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

$$MAE = \frac{1}{n} \sum_{i=1}^{i=n} |y_i - f_{Hybrid}(x_i)| \tag{6.2}$$

It penalizes all the errors evenly. That's a unique property of MAE. The MAE doesn't bifurcate 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 the MSE over the MAE and other error measures, the mean square error (MSE) has been used to illustrate the ideas in the subsequent sections.

6.2 Universal Metrics of a Model

If we disregard the system-specific evaluation metrics, up until now, all the models were only evaluated for accuracy within the dataset. Holistic evaluation has never been a priority. We propose a five-faceted generic evaluation framework for the wholesome evaluation of a model of the CPS. It includes the accuracy of the model within and beyond the range of the training data. As noise is inseparable from the data, how to evaluate the robustness of the model to noise? How sensitive is the model to data? 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. They are as follows:

6.2.1 Interpolability

The model learns a continuous function f_{Hybrid} from the sparse dataset S. So, the complete manifold $\mathcal{M}$ derived from the sparsely populated dataset may not exactly represent the subspace $\mathcal{S}$ of the whole space $\mathbb{S}$ that is actually carved out by the system. Mathematically, $S \mapsto \mathcal{M} \approx \mathcal{S} \subset \mathbb{S}$. 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*. Mathematically, the interpolability measure can be defined as

$$MSE_{Int} = \frac{1}{n}\sum_{i=1}^{i=n}[y_i - f_{Hybrid}(x_i)]^2, x_i \in Range(S) \tag{6.3}$$

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 interpolabilty metric. That's why the earlier deep learning architectures failed drastically against unseen datasets until the development of the ImageNet database [89, 90] which included data from diverse categories. Still, they are vulnerable to fail against unseen data. Their success can be attributed to the diverse data they have been trained upon which ultimately reduced the type of data that they can ever be exposed to beyond their training range.

6.2.2 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 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*. Mathematically, the extrapolability can be defined as

$$MSE_{Ext} = \frac{1}{n}\sum_{i=1}^{i=n}[y_i - f_{Hybrid}(x_i)]^2, x_i \notin Range(S) \tag{6.4}$$

The interpolability metric and the extrapolability metric are mutually exclusive. A model with good interpolabilty and extrapolability can be said to be complete as they can perform well within and beyond the boundaries of the dataset. The hybrid models have portrayed excellent extrapolability [146, 152, 168, 179, 181, 197]. However, the extrapolability of a model of a CPS is quite difficult to evaluate as data collection beyond the range of the current training data could be challenging.

6.2.3 Sensitivity to the size of data

As the hybrid models incorporate ML models, they become sensitive to the size of data. Prior work has shown that they are comparatively less sensitive than pure data-driven models [146]. The sensitivity to the size of data is an important metric because of the cost of the data acquisition process in case of CPS applications like manufacturing, healthcare etc. As data collection needs domain knowledge, it cannot be crowd-sourced. That makes the data collection process expensive. In such cases, the model which is least sensitive to data is desirable. The sensitivity to the size of data can be measured directly from the MSE. It can be defined as

$$MSE_{Data} = \frac{1}{n}\sum_{i=1}^{i=n}[y_i - f_{Hybrid}(x_i)]^2 \tag{6.5}$$

6.2.4 Robustness to Noisy Data

As hybrid models equip ML models, they are not insensitive to noisy data. So, the model must be evaluated against noisy inputs. Noisy inputs tend to affect the local behavior of the model. This is a crucial metric as noisy data is inherent to the system. Chaos of the environment could a factor behind noisy data. Artificial noise 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 $\mathcal{N}$ defined as $\mathcal{N} \sim (\mu, \sigma^2)$ where μ and σ^2 are the mean and variance of the noise respectively, the MSE is defined as:

$$MSE_{\mathcal{N}} = \frac{1}{n}\sum_{i=1}^{i=n}[y_i - f_{Hybrid}(x_i + \mathcal{N})]^2 \tag{6.6}$$

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. Hybrid models have been proven to be more robust to noise than pure MB and ML methods [146].

6.2.5 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 caused mostly by the complexity of the model must also be included. Model complexity translates into time of processing and memory requirements. A complex computational model will require higher memory and time for processing. The computation time is estimated based on the number of floating point operations (FLOPs). As CPS are live systems, complex computational models can delay the response time of the CPS. Thereby compromising the dynamic interaction of the CPS with the environment. The memory is required for storing the parameters of the model e.g. weight matrices, bias matrices, physics parameters etc. Bigger, deeper networks and complex physics models tend to need more parameters. That will require 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. The FLOPs and memory must be monitored.

Chapter 7

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 modern ML subverts several shortcomings of the MB as discussed in section 4. Due to this, there has been a lot of interest in the hybrid modeling domain. Despite the significant thrust for 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. Deciding the mode of combination for maximum effectiveness is a challenge which was the prime focus of our work. Additionally, the hybrid models also face the hurdles of ML. Other challenges which we did not cover but are of significant importance for accelerating the progress of hybrid modeling are discussed below.

7.1 Model Selection based on Hybrid Models

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 case of hybrid models. Appropriate MB 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 and a combination of them. Several MB 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.

7.2 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 [89, 90] and the inverted pendulum [198, 199] are the benchmark problems for object detection and reinforcement learning tasks respectively. But, no benchmark CPS 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 [91] to CIFAR-10 [92] to ImageNet 2009 [89] and 2015 [90]. 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 evolve with the expertise. Also, as CPS have diverse applications, it may require the formulation of unique benchmark problems for each domain.

7.3 Regression Problems with low amount of data

The deep learning based techniques have seen unprecedented success in case of supervised learning. One of the factors behind the success is the availability of enormous amounts of data. On the other hand, MB has succeeded without the need of any data. They represent two ends of the spectrum. In the spectrum from full physics to no physics and from zero data to big data, there lies an area with limited data and limited physics. That's where most of the CPS belong. For modeling a system with limited data and limited physics, both ML and MB must be combined. For modeling such systems, individually they may be incapable but together they can be invincible. The prior works [146, 152, 168, 179, 181] 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, their work has just initiated the excellence of hybrid models for limited data over traditional methods. Their domain of application is still limited to simple systems. They have not been exclusively tested in CPS applications. Hence, exclusive and comprehensive evaluation of hybrid models against CPS with limited data is required.

7.4 Learning with Less Labels

Supervised learning, which uses labeled data, has experienced unanticipated success with deep learning. At the same time, unsupervised learning, which uses data without labels, is still in a nascent stage. An intermediary which is referred as semi-supervised learning uses less amount of labeled data along with huge amount of unlabeled data. Semi-supervised learning is the pilot test for unsupervised learning. CPS deployed in open environments cannot be supervised completely. Moreover, as CPS find applications in diverse domains, labeling may require domain specific expertise. Thus, they come under the scope of semi-supervised learning. So, they can utilize techniques like active learning [200] and semi-supervised learning [201] . Hybrid learning techniques are yet to step into these areas. As a beginner's step, the motion of a ball was predicted in a video using a constrained hybrid model [170]. Success of unsupervised learning in CPS can truly make the CPS ubiquitous. In addition to this, the error metrics defined in chapter 6 are for supervised learning where the ground truth labels are available. For data without ground truth labels, similar metrics for unsupervised learning [202] shall be devised.

7.5 Intelligent Collection of data 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 behaviour of the system. The hybrid models have been proven to be efficient at capturing the system behavior with relatively smaller datasets [146, 152, 168, 179, 181]. 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 we are actually capturing noise is unwise. But, as no clear demarcation exist between local and global behavior, it is difficult to collect data intelligently to capture the global behavior.

Moreover, 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 stagnation of the accuracy of the model is imprudent. So, the accuracy of the model shall be evaluated in a running basis. And the data collection from the same subspace shall be stopped once the accuracy plateaus. It can be collected from newer subspaces. So, new data shall be collected with the intent of increasing running-average MSE_{Int} and the model shall be trained with the goal of reducing the running-average MSE_{Int}. The running average is equivalent to one-dimensional convolution with average pooling. To summarize, the process of data collection and model training shall follow a non-cooperative game theoretic approach.

7.6 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 [203]. 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 [204, 205] 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 [206] 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 for heading further. Among the geometric deep learning techniques, structured graph CNNs [205] and spectral clustering [207] techniques have already been utilized for CPS. Other techniques like spectral CNNs [208, 209], graph CNNs [210, 211], geodesic CNNs [212], manifold learning should be explored for CPS applications. Geometric deep learning techniques can be instrumental for processing big data [213] in a cyber-physical setup.

7.7 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 of the cases, misclassifying the 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 [214]. However, their performance at mitigating the impact of class imbalance is not as expected and hence debatable [214, 215, 216]. Intelligent data collection can aide in mitigating the impact of imbalanced data. The metrics should also be modified aptly for monitoring the impact of data imbalance [216]. However, We still have a long way to go for correctly addressing imbalanced data and data at tail.

Chapter 8

Conclusion

This is a diligent attempt at disseminating the collective intelligence of hybrid models (MB + ML) in an organized manner. To the best of our knowledge, no such attempt has been made yet. 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 mitigate the limitations of ML models arising due to the time resolution of the data acquisition system. The ML can suppress the impact of heterogeneity, time asynchrony, delay and loss in the network infrastructure of the MB. Overall, the combination is a win-win situation. Moreover, we proposed a five-faceted generic evaluation framework for all-round evaluation of a model. It scales beyond the conventional metrics of accuracy within the space. It also monitors the model's complexity, robustness to noisy data and sensitivity to the size of data for optimal performance. Finally, the challenges faced by hybrid models which deter the unbridled growth of hybrid models are briefed. We hope the current work to become a stepping stone at the organized dispersal and growth of hybrid models.

Bibliography

[1] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[2] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[3] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

[4] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. 1999.

[5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[6] Carl Doersch. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*, 2016.

[7] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[8] Sandro Skansi. *Introduction to Deep Learning: from logical calculus to artificial intelligence*. Springer, 2018.

[9] Edward Ashford Lee and Sanjit A Seshia. *Introduction to embedded systems: A cyber-physical systems approach*. Mit Press, 2016.

[10] Radhakisan Baheti and Helen Gill. Cyber-physical systems. *The impact of control technology*, 12(1):161–166, 2011.

[11] Ragunathan Rajkumar, Insup Lee, Lui Sha, and John Stankovic. Cyber-physical systems: the next computing revolution. In *Design Automation Conference*, pages 731–736. IEEE, 2010.

[12] Jay Lee, Behrad Bagheri, and Hung-An Kao. A cyber-physical systems architecture for industry 4.0-based manufacturing systems. *Manufacturing letters*, 3:18–23, 2015.

[13] Birgit Vogel-Heuser, Jay Lee, and Paulo Leitão. Agents enabling cyber-physical production systems. *at-Automatisierungstechnik*, 63(10):777–789, 2015.

[14] Anuj Karpatne, Gowtham Atluri, James H Faghmous, Michael Steinbach, Arindam Banerjee, Auroop Ganguly, Shashi Shekhar, Nagiza Samatova, and Vipin Kumar. Theory-guided data science: A new paradigm for scientific discovery from data. *IEEE Transactions on Knowledge and Data Engineering*, 29(10):2318–2331, 2017.

[15] Insup Lee. Cyber physical systems: The next computing revolution. *UPenn. Internet: https://www.seas.upenn.edu/ lee/10cis541/lecs/lec-CPS-1x2.pdf [CIS 541 Spring 2010]*, 2010.

[16] Shah Ahsanul Haque, Syed Mahfuzul Aziz, and Mustafizur Rahman. Review of cyber-physical system in healthcare. *international journal of distributed sensor networks*, 10(4):217415, 2014.

[17] Hausi A. Muller. The rise of intelligent cyber-physical systems. *Computer: https://www.computer.org/csdl/magazine/co/2017/12/mco2017120007/13rRUytF44K*, 50:7–9, 2017.

[18] Phil Goldstein. How do feds use and secure cyber-physical systems? *https://fedtechmagazine.com/article/2018/01/how-do-feds-use-and-secure-cyber-physical-systems*, 2018.

[19] Roberto Sabella. Cyber physical systems for industry 4.0. *https://www.ericsson.com/en/blog/2018/10/cyber-physical-systems-for-industry-4.0*, 2018.

[20] Jianhua Shi, Jiafu Wan, Hehua Yan, and Hui Suo. A survey of cyber-physical systems. In *2011 international conference on wireless communications and signal processing (WCSP)*, pages 1–6. IEEE, 2011.

[21] Helen Gill. Nsf perspective and status on cyber-physical systems. *Austin. Internet: http://varma. ece. cmu. edu/CPS/Presentations/gill. pdf [zuletzt aufgesucht am 1.04. 2015]*, 2006.

[22] Networking, Information Technology Research, and Cyber-Physical Systems (CPS) Interagency Working Group (IWG) Development. Cyber physical systems.

[23] Farhad Mehdipour, Krishna Chaitanya Nunna, and Kazuaki J Murakami. A smart cyber-physical systems-based solution for pest control (work in progress). In *2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*, pages 1248–1253. IEEE, 2013.

[24] Farhad Mehdipour. Smart field monitoring: An application of cyber-physical systems in agriculture (work in progress). In *2014 IIAI 3rd International Conference on Advanced Applied Informatics*, pages 181–184. IEEE, 2014.

[25] Simona Iuliana Caramihai and Ioan Dumitrache. Agricultural enterprise as a complex system: A cyber physical systems approach. In *2015 20th International Conference on Control Systems and Computer Science*, pages 659–664. IEEE, 2015.

[26] Ciprian-Radu Rad, Olimpiu Hancu, Ioana-Alexandra Takacs, and Gheorghe Olteanu. Smart monitoring of potato crop: a cyber-physical system architecture model in the field of precision agriculture. *Agriculture and Agricultural Science Procedia*, 6:73–79, 2015.

[27] Chi-Un Lei, Hai-Ning Liang, and Ka Lok Man. Building a smart laboratory environment at a university via a cyber-physical system. In *Proceedings of 2013 IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE)*, pages 243–247. IEEE, 2013.

[28] Chi-Un Lei, Kaiyu Wan, and Ka Lok Man. Developing a smart learning environment in universities via cyber-physical systems. *Procedia Computer Science*, 17:583–585, 2013.

[29] Vincent Ricquebourg, David Menga, David Durand, Bruno Marhic, Laurent Delahoche, and Christophe Loge. The smart home concept: our immediate future. In *2006 1st IEEE international conference on e-learning in industrial electronics*, pages 23–28. IEEE, 2006.

[30] Francesco Basile, Pasquale Chiacchio, Jolanda Coppola, and Diego Gerbasio. Automated warehouse systems: A cyber-physical system perspective. In *2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA)*, pages 1–4. IEEE, 2015.

[31] Qaisar Shafi. Cyber physical systems security: A brief survey. In *2012 12th International Conference on Computational Science and Its Applications*, pages 146–150. IEEE, 2012.

[32] Peiyuan Dong, Yue Han, Xiaobo Guo, and Feng Xie. A systematic review of studies on cyber physical system security. *International Journal of Security and Its Applications*, 9(1):155–164, 2015.

[33] Siddharth Sridhar, Adam Hahn, and Manimaran Govindarasu. Cyber–physical system security for the electric power grid. *Proceedings of the IEEE*, 100(1):210–224, 2011.

[34] Yining Li, Jing Wu, and Shaoyuan Li. State estimation for distributed cyber-physical power systems under data attacks. *International Journal of Modelling, Identification and Control*, 26(4):317–323, 2016.

[35] Anthony P Hamins, Nelson P Bryner, Albert W Jones, Galen H Koepke, et al. Research roadmap for smart fire fighting. Technical report, 2015.

[36] Vania V Estrela, Jude Hemanth, Osamu Saotome, Edwiges GH Grata, and Daniel RF Izario. Emergency response cyber-physical system for flood prevention with sustainable electronics. In *Brazilian Technology Symposium*, pages 319–328. Springer, 2017.

[37] Erol Gelenbe and Fang-Jing Wu. Future research on cyber-physical emergency management systems. *Future Internet*, 5(3):336–354, 2013.

[38] Vania Estrela, Osamu Saotome, Hermes Loschi, Jude Hemanth, Willian Farfan, Jenice Aroma, Chandran Saravanan, and Edwiges Grata. Emergency response cyber-physical framework for landslide avoidance with sustainable electronics. *Technologies*, 6(2):42, 2018.

[39] Yu-Ting Li, Mithun Jacob, George Akingba, and Juan P Wachs. A cyber-physical management system for delivering and monitoring surgical instruments in the or. *Surgical innovation*, 20(4):377–384, 2013.

[40] Daniel Sonntag, Sonja Zillner, Samarjit Chakraborty, Andras Lorincz, Esko Strommer, and Luciano Serafini. The medical cyber-physical systems activity at eit: A look under the hood. In *2014 IEEE 27th International symposium on computer-based medical systems*, pages 351–356. IEEE, 2014.

[41] Christos G Cassandras. Smart cities as cyber-physical social systems. *Engineering*, 2(2):156–158, 2016.

[42] Dong Jin, Christopher Hannon, Zhiyi Li, Pablo Cortes, Srinivasan Ramaraju, Patrick Burgess, Nathan Buch, and Mohammad Shahidehpour. Smart street lighting system: A platform for innovative smart city applications and a new frontier for cyber-security. *The Electricity Journal*, 29(10):28–35, 2016.

[43] Jiafu Wan, Hehua Yan, Di Li, Keliang Zhou, and Lu Zeng. Cyber-physical systems for optimal energy management scheme of autonomous electric vehicle. *The Computer Journal*, 56(8):947–956, 2013.

[44] Gang Xiong, Fenghua Zhu, Xiwei Liu, Xisong Dong, Wuling Huang, Songhang Chen, and Kai Zhao. Cyber-physical-social system in intelligent transportation. *IEEE/CAA Journal of Automatica Sinica*, 2(3):320–333, 2015.

[45] Teodora Sanislav, George Mois, Silviu Folea, Liviu Miclea, Giulio Gambardella, and Paolo Prinetto. A cloud-based cyber-physical system for environmental monitoring. In *2014 3rd Mediterranean Conference on Embedded Computing (MECO)*, pages 6–9. IEEE, 2014.

[46] Chao Liu and Pingyu Jiang. A cyber-physical system architecture in shop floor for intelligent manufacturing. *Procedia Cirp*, 56:372–377, 2016.

[47] Behrad Bagheri, Shanhu Yang, Hung-An Kao, and Jay Lee. Cyber-physical systems architecture for self-aware machines in industry 4.0 environment. *IFAC-PapersOnLine*, 48(3):1622–1627, 2015.

[48] Radu F Babiceanu and Remzi Seker. Big data and virtualization for manufacturing cyber-physical systems: A survey of the current status and future outlook. *Computers in Industry*, 81:128–137, 2016.

[49] Jay Lee, Hossein Davari Ardakani, Shanhu Yang, and Behrad Bagheri. Industrial big data analytics and cyber-physical systems for future maintenance & service innovation. *Procedia Cirp*, 38:3–7, 2015.

[50] Hong Chen. Applications of cyber-physical system: a literature review. *Journal of Industrial Integration and Management*, 2(03):1750012, 2017.

[51] Siddhartha Kumar Khaitan and James D McCalley. Cyber physical system approach for design of power grids: A survey. In *2013 IEEE Power & Energy Society General Meeting*, pages 1–5. IEEE, 2013.

[52] ChuanTao Yin, Zhang Xiong, Hui Chen, JingYuan Wang, Daven Cooper, and Bertrand David. A literature survey on smart cities. *Science China Information Sciences*, 58(10):1–18, 2015.

[53] Dongyao Jia, Kejie Lu, Jianping Wang, Xiang Zhang, and Xuemin Shen. A survey on platoon-based vehicular cyber-physical systems. *IEEE communications surveys & tutorials*, 18(1):263–284, 2015.

[54] Thomas H. Davenport and Julia Kirby. Beyond automation. *https://ibm.ent.box.com/s/5fhhabp83jafccemsbmyvlnq2ocbw188*, 2015.

[55] George H Mealy. A method for synthesizing sequential circuits. *The Bell System Technical Journal*, 34(5):1045–1079, 1955.

[56] Edward F Moore. Gedanken-experiments on sequential machines. *Automata studies*, 34:129–153, 1956.

[57] Jing Lin, Sahra Sedigh, and Ann Miller. Modeling cyber-physical systems with semantic agents. In *2010 IEEE 34th Annual Computer Software and Applications Conference Workshops*, pages 13–18. IEEE, 2010.

[58] Jing Lin, Sahra Sedigh, and Ann Miller. A semantic agent framework for cyber-physical systems. In *Semantic Agent Systems*, pages 189–213. Springer, 2011.

[59] Paulo Leitao, Stamatis Karnouskos, Luis Ribeiro, Jay Lee, Thomas Strasser, and Armando W Colombo. Smart agents in industrial cyber–physical systems. *Proceedings of the IEEE*, 104(5):1086–1101, 2016.

[60] Eric Bonabeau. Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the national academy of sciences*, 99(suppl 3):7280–7287, 2002.

[61] Budhitama Subagdja. Agent-based modeling for developing pervasive persuasive systems. In *2015 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, pages 23–28. IEEE, 2015.

[62] Aaron Hector and V Lakshmi Narasimhan. A new classification scheme for software agents. In *Third International Conference on Information Technology and Applications (ICITA'05)*, volume 1, pages 191–196. IEEE, 2005.

[63] Ali Aliyuda. Towards the design of cyber-physical system via multi-agent system technology. *International Journal of Scientific Engineering Research*, 7(10):155–161, 2016.

[64] Peter Pederson, Danile Dudenhoeffer, Steven Hartley, and May Permann. Critical infrastructure interdependency modeling: a survey of us and international research. *Idaho National Laboratory*, 25:27, 2006.

[65] Aleksandar Lj Juloski, Siep Weiland, and WPMH Heemels. A bayesian approach to identification of hybrid systems. *IEEE Transactions on Automatic Control*, 50(10):1520–1533, 2005.

[66] René Vidal, Stefano Soatto, Yi Ma, and Shankar Sastry. An algebraic geometric approach to the identification of a class of linear hybrid systems. In *42nd IEEE International Conference on Decision and Control (IEEE Cat. No. 03CH37475)*, volume 1, pages 167–172. IEEE, 2003.

[67] Alberto Bemporad, Andrea Garulli, Simone Paoletti, and Antonio Vicino. A bounded-error approach to piecewise affine system identification. *IEEE Transactions on Automatic Control*, 50(10):1567–1580, 2005.

[68] Necmiye Ozay, Mario Sznaier, Constantino M Lagoa, and Octavia I Camps. A sparsification approach to set membership identification of switched affine systems. *IEEE Transactions on Automatic Control*, 57(3):634–648, 2011.

[69] Ye Yuan, Xiuchuan Tang, Wei Zhou, Wei Pan, Xiuting Li, Hai-Tao Zhang, Han Ding, and Jorge Goncalves. Data driven discovery of cyber physical systems. *Nature Communications*, 10(1):4894, 2019.

[70] Alberto Bemporad, Valentina Breschi, and Dario Piga. Piecewise affine regression via recursive multiple least squares and multicategory discrimination. *Automatica, accepted for publication*, 2016.

[71] Daniel L Ly and Hod Lipson. Learning symbolic representations of hybrid dynamical systems. *Journal of Machine Learning Research*, 13(Dec):3585–3618, 2012.

[72] Gianluigi Pillonetto. A new kernel-based approach to hybrid system identification. *Automatica*, 70:21–31, 2016.

[73] Simone Paoletti, Aleksandar Lj Juloski, Giancarlo Ferrari-Trecate, and René Vidal. Identification of hybrid systems a tutorial. *European journal of control*, 13(2-3):242–260, 2007.

[74] Hayato Nakada, Kiyotsugu Takaba, and Tohru Katayama. Identification of piecewise affine systems based on statistical clustering technique. *Automatica*, 41(5):905–913, 2005.

[75] Richard HR Hahnloser, Rahul Sarpeshkar, Misha A Mahowald, Rodney J Douglas, and H Sebastian Seung. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*, 405(6789):947, 2000.

[76] George E Dahl, Tara N Sainath, and Geoffrey E Hinton. Improving deep neural networks for lvcsr using rectified linear units and dropout. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 8609–8613. IEEE, 2013.

[77] Matthew D Zeiler, M Ranzato, Rajat Monga, Min Mao, Kun Yang, Quoc Viet Le, Patrick Nguyen, Alan Senior, Vincent Vanhoucke, Jeffrey Dean, et al. On rectified linear units for speech processing. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3517–3521. IEEE, 2013.

[78] Jun Han and Claudio Moraga. The influence of the sigmoid function parameters on the speed of backpropagation learning. In *International Workshop on Artificial Neural Networks*, pages 195–201. Springer, 1995.

[79] Tommaso Dreossi, Alexandre Donzé, and Sanjit A Seshia. Compositional falsification of cyber-physical systems with machine learning components. *Journal of Automated Reasoning*, 63(4):1031–1053, 2019.

[80] Chathurika S Wickramasinghe, Daniel L Marino, Kasun Amarasinghe, and Milos Manic. Generalization of deep learning for cyber-physical system security: A survey. In *IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society*, pages 745–751. IEEE, 2018.

[81] André Platzer. The logical path to autonomous cyber-physical systems. In *International Conference on Quantitative Evaluation of Systems*, pages 25–33. Springer, 2019.

[82] Patricia Derler, Edward A Lee, and Alberto Sangiovanni Vincentelli. Modeling cyber–physical systems. *Proceedings of the IEEE*, 100(1):13–28, 2011.

[83] Edward A Lee. Fundamental limits of cyber-physical systems modeling. *ACM Transactions on Cyber-Physical Systems*, 1(1):3, 2017.

[84] Aaron D Ames, Alessandro Abate, and Shankar Sastry. Sufficient conditions for the existence of zeno behavior. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 696–701. IEEE, 2005.

[85] Lothar Thiele and Pratyush Kumar. Can real-time systems be chaotic? In *2015 International Conference on Embedded Software (EMSOFT)*, pages 21–30. IEEE, 2015.

[86] Fei-Yue Wang, Xiao Wang, Lingxi Li, and Li Li. Steps toward parallel intelligence. *IEEE/CAA Journal of Automatica Sinica*, 3(4):345–348, 2016.

[87] William E Wilkins. The concept of a self-fulfilling prophecy. *Sociology of Education*, pages 175–183, 1976.

[88] Daya Krishna. " the self-fulfilling prophecy" and the nature of society. *American Sociological Review*, 36(6):1104–1107, 1971.

[89] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[90] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[91] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, pages 178–178. IEEE, 2004.

[92] Alex Krizhevsky and Geoff Hinton. Convolutional deep belief networks on cifar-10. *Unpublished manuscript*, 40(7):1–9, 2010.

[93] Cade Metz. How google's ai viewed the move no human could understand, 2016.

[94] Stavros Tripakis. Data-driven and model-based design. In *2018 IEEE Industrial Cyber-Physical Systems (ICPS)*, pages 103–108. IEEE, 2018.

[95] Asa Ben-Hur, David Horn, Hava T Siegelmann, and Vladimir Vapnik. Support vector clustering. *Journal of machine learning research*, 2(Dec):125–137, 2001.

[96] Hyeran Byun and Seong-Whan Lee. Applications of support vector machines for pattern recognition: A survey. In *International Workshop on Support Vector Machines*, pages 213–236. Springer, 2002.

[97] Ashis Pradhan. Support vector machine-a survey. *International Journal of Emerging Technology and Advanced Engineering*, 2(8):82–85, 2012.

[98] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[99] Antanas Verikas, Adas Gelzinis, and Marija Bacauskiene. Mining data with random forests: A survey and results of new tests. *Pattern recognition*, 44(2):330–349, 2011.

[100] Salvador García, Sergio Ramírez-Gallego, Julián Luengo, José Manuel Benítez, and Francisco Herrera. Big data preprocessing: methods and prospects. *Big Data Analytics*, 1(1):9, 2016.

[101] Oliver Faust, Yuki Hagiwara, Tan Jen Hong, Oh Shu Lih, and U Rajendra Acharya. Deep learning for healthcare applications based on physiological signals: A review. *Computer methods and programs in biomedicine*, 161:1–13, 2018.

[102] Yannick Roy, Hubert Banville, Isabela Albuquerque, Alexandre Gramfort, Tiago H Falk, and Jocelyn Faubert. Deep learning-based electroencephalography analysis: a systematic review. *Journal of neural engineering*, 2019.

[103] Wenchang Zhang, Chuanqi Tan, Fuchun Sun, Hang Wu, and Bo Zhang. A review of eeg-based brain-computer interface systems design. *Brain Science Advances*, 4(2):156–167, 2018.

[104] Zhiqiang Zhang, Yi Zhao, Xiangke Liao, Wenqiang Shi, Kenli Li, Quan Zou, and Shaoliang Peng. Deep learning in omics: a survey and guideline. *Briefings in functional genomics*, 18(1):41–57, 2018.

[105] Funa Zhou, Yulin Gao, and Chenglin Wen. A novel multimode fault classification method based on deep learning. *Journal of Control Science and Engineering*, 2017, 2017.

[106] Rui Zhao, Ruqiang Yan, Zhenghua Chen, Kezhi Mao, Peng Wang, and Robert X Gao. Deep learning and its applications to machine health monitoring. *Mechanical Systems and Signal Processing*, 115:213–237, 2019.

[107] Yaguo Lei, Jing Lin, Zhengjia He, and Ming J Zuo. A review on empirical mode decomposition in fault diagnosis of rotating machinery. *Mechanical systems and signal processing*, 35(1-2):108–126, 2013.

[108] Ruqiang Yan, Robert X Gao, and Xuefeng Chen. Wavelets for fault diagnosis of rotary machines: A review with applications. *Signal processing*, 96:1–15, 2014.

[109] ZK Peng, W Tse Peter, and FL Chu. A comparison study of improved hilbert–huang transform and wavelet transform: application to fault diagnosis for rolling bearing. *Mechanical systems and signal processing*, 19(5):974–988, 2005.

[110] Boyuan Yang, Ruonan Liu, and Xuefeng Chen. Fault diagnosis for a wind turbine generator bearing via sparse representation and shift-invariant k-svd. *IEEE Transactions on Industrial Informatics*, 13(3):1321–1331, 2017.

[111] Ruonan Liu, Boyuan Yang, Enrico Zio, and Xuefeng Chen. Artificial intelligence for fault diagnosis of rotating machinery: A review. *Mechanical Systems and Signal Processing*, 108:33–47, 2018.

[112] Mohammadkazem Sadoughi and Chao Hu. Physics-based convolutional neural network for fault diagnosis of rolling element bearings. *IEEE Sensors Journal*, 19(11):4181–4192, 2019.

[113] Mohammadkazem Sadoughi and Chao Hu. A physics-based deep learning approach for fault diagnosis of rotating machinery. In *IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society*, pages 5919–5923. IEEE, 2018.

[114] Yumei Qi, Wei You, Changqing Shen, Xingxing Jiang, Weiguo Huang, and Zhongkui Zhu. Hierarchical diagnosis network based on sparse deep neural networks and its application in bearing fault diagnosis. In *2017 Prognostics and System Health Management Conference (PHM-Harbin)*, pages 1–7. IEEE, 2017.

[115] Jihoon Hong, Qiushi Wang, Xueheng Qiu, and Hian Leng Chan. Remaining useful life prediction using time-frequency feature and multiple recurrent neural networks. In *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 916–923. IEEE, 2019.

[116] Guorong Xiao, Akhil Garg, Dicheng Chen, Dazhi Jiang, Wanneng Shu, and Xuemiao Xu. Ahe detection with a hybrid intelligence model in smart healthcare. *IEEE Access*, 7:37360–37370, 2019.

[117] C Venkatesan, P Karthigaikumar, Anand Paul, S Satheeskumaran, and RJIA Kumar. Ecg signal preprocessing and svm classifier-based abnormality detection in remote healthcare applications. *IEEE Access*, 6:9767–9773, 2018.

[118] Aapo Hyvärinen, Pavan Ramkumar, Lauri Parkkonen, and Riitta Hari. Independent component analysis of short-time fourier transforms for spontaneous eeg/meg analysis. *NeuroImage*, 49(1):257–271, 2010.

[119] Jatin Bedi and Durga Toshniwal. Empirical mode decomposition based deep learning for electricity demand forecasting. *IEEE Access*, 6:49144–49156, 2018.

[120] Wen-chuan Wang, Kwok-wing Chau, Lin Qiu, and Yang-bo Chen. Improving forecasting accuracy of medium and long-term runoff using artificial neural network based on eemd decomposition. *Environmental research*, 139:46–54, 2015.

[121] Cong Ma and Yangyang Li. Improving forecasting accuracy of annual runoff time series using rbfn based on eemd decomposition. *DEStech Transactions on Engineering and Technology Research*, (emme), 2016.

[122] Xike Zhang, Qiuwen Zhang, Gui Zhang, Zhiping Nie, and Zifan Gui. A hybrid model for annual runoff time series forecasting using elman neural network with ensemble empirical mode decomposition. *Water*, 10(4):416, 2018.

[123] Mingfei Niu, Kai Gan, Shaolong Sun, and Fengying Li. Application of decomposition-ensemble learning paradigm with phase space reconstruction for day-ahead pm2. 5 concentration forecasting. *Journal of environmental management*, 196:110–118, 2017.

[124] Feng Jia, Yaguo Lei, Jing Lin, Xin Zhou, and Na Lu. Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data. *Mechanical Systems and Signal Processing*, 72:303–315, 2016.

[125] Zhiqiang Chen, Shengcai Deng, Xudong Chen, Chuan Li, René-Vinicio Sanchez, and Huafeng Qin. Deep neural networks-based rolling bearing fault diagnosis. *Microelectronics Reliability*, 75:327–333, 2017.

[126] Jiedi Sun, Changhong Yan, and Jiangtao Wen. Intelligent bearing fault diagnosis method combining compressed data acquisition and deep learning. *IEEE Transactions on Instrumentation and Measurement*, 67(1):185–195, 2017.

[127] Tan Junbo, Lu Weining, An Juneng, and Wan Xueqian. Fault diagnosis method study in roller bearing based on wavelet transform and stacked auto-encoder. In *The 27th Chinese Control and Decision Conference (2015 CCDC)*, pages 4608–4613. IEEE, 2015.

[128] Peng Wang, Ruqiang Yan, Robert X Gao, et al. Virtualization and deep recognition for system fault classification. *Journal of Manufacturing Systems*, 44:310–316, 2017.

[129] Xiaoxi Ding and Qingbo He. Energy-fluctuated multiscale feature learning with deep convnet for intelligent spindle bearing fault diagnosis. *IEEE Transactions on Instrumentation and Measurement*, 66(8):1926–1935, 2017.

[130] Jinjiang Wang, Junfei Zhuang, Lixiang Duan, and Weidong Cheng. A multi-scale convolution neural network for featureless fault diagnosis. In *2016 International Symposium on Flexible Automation (ISFA)*, pages 65–70. IEEE, 2016.

[131] Faisal AlThobiani, Andrew Ball, et al. An approach to fault diagnosis of reciprocating compressor valves using teager–kaiser energy operator and deep belief networks. *Expert Systems with Applications*, 41(9):4113–4122, 2014.

[132] Jinjiang Wang, Yulin Ma, Laibin Zhang, Robert X Gao, and Dazhong Wu. Deep learning for smart manufacturing: Methods and applications. *Journal of Manufacturing Systems*, 48:144–156, 2018.

[133] Jing Tian, Carlos Morillo, Michael H Azarian, and Michael Pecht. Motor bearing fault detection using spectral kurtosis-based feature extraction coupled with k-nearest neighbor distance analysis. *IEEE Transactions on Industrial Electronics*, 63(3):1793–1803, 2015.

[134] David Abboud, J Antoni, S Sieg-Zieba, and M Eltabach. Envelope analysis of rotating machine vibrations in variable speed conditions: A comprehensive treatment. *Mechanical Systems and Signal Processing*, 84:200–226, 2017.

[135] Zhaohua Wu and Norden E Huang. Ensemble empirical mode decomposition: a noise-assisted data analysis method. *Advances in adaptive data analysis*, 1(01):1–41, 2009.

[136] Norbert A Agana, Emmanuel Oleka, Gabriel Awogbami, and Abdollah Homaifar. Short-term load forecasting based on a hybrid deep learning model. In *SoutheastCon 2018*, pages 1–6. IEEE, 2018.

[137] Seetaram Maurya, Vikas Singh, and Nishchal K Verma. Condition monitoring of machines using fused features from emd based local energy with dnn. *IEEE Sensors Journal*, 2019.

[138] Adam Harasimowicz. Comparison of data preprocessing methods and the impact on auto-encoders performance in activity recognition domain. *ICT Young 2014*, page 6, 2014.

[139] Davide Figo, Pedro C Diniz, Diogo R Ferreira, and João M Cardoso. Preprocessing techniques for context recognition from accelerometer data. *Personal and Ubiquitous Computing*, 14(7):645–662, 2010.

[140] Salvador García, Julián Luengo, and Francisco Herrera. *Data preprocessing in data mining*. Springer, 2015.

[141] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. Neural architecture search: A survey. *arXiv preprint arXiv:1808.05377*, 2018.

[142] Sirui Xie, Hehui Zheng, Chunxiao Liu, and Liang Lin. Snas: stochastic neural architecture search. *arXiv preprint arXiv:1812.09926*, 2018.

[143] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.

[144] Chenxi Liu, Barret Zoph, Maxim Neumann, Jonathon Shlens, Wei Hua, Li-Jia Li, Li Fei-Fei, Alan Yuille, Jonathan Huang, and Kevin Murphy. Progressive neural architecture search. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 19–34, 2018.

[145] Anuj Karpatne, William Watkins, Jordan Read, and Vipin Kumar. Physics-guided neural networks (pgnn): An application in lake temperature modeling. *arXiv preprint arXiv:1710.11431*, 2017.

[146] Shubhendu Singh, Ruoyu Yang, Amir Behjat, Rahul Rai, Souma Chowdhury, and Ion Matei. Pi-lstm: Physics-infused long short-term memory network. *IEEE International Conference on Machine Learning and Applications*, 2019.

[147] Yunhao Ba, Rui Chen, Yiqin Wang, Lei Yan, Boxin Shi, and Achuta Kadambi. Physics-based neural networks for shape from polarization. *arXiv preprint arXiv:1903.10210*, 2019.

[148] Nima Mohajerin and Steven L Waslander. Multistep prediction of dynamic systems with recurrent neural networks. *IEEE transactions on neural networks and learning systems*, 2019.

[149] Anurag Ajay, Jiajun Wu, Nima Fazeli, Maria Bauza, Leslie P Kaelbling, Joshua B Tenenbaum, and Alberto Rodriguez. Augmenting physical simulators with stochastic neural networks: Case study of planar pushing and bouncing. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3066–3073. IEEE, 2018.

[150] Andy Zeng, Shuran Song, Johnny Lee, Alberto Rodriguez, and Thomas Funkhouser. Tossingbot: Learning to throw arbitrary objects with residual physics. *arXiv preprint arXiv:1903.11239*, 2019.

[151] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[152] Michael Lutter, Christian Ritter, and Jan Peters. Deep lagrangian networks: Using physics as model prior for deep learning. *arXiv preprint arXiv:1907.04490*, 2019.

[153] Prashanth Pillai, Anshul Kaushik, Shivanand Bhavikatti, Arjun Roy, and Virendra Kumar. A hybrid approach for fusing physics and data for failure prediction. *International Journal of Prognostics and Health Management*, 7(025):1–12, 2016.

[154] Chih-Chieh Young, Wen-Cheng Liu, and Ming-Chang Wu. A physically based and machine learning hybrid approach for accurate rainfall-runoff modeling during extreme typhoon events. *Applied Soft Computing*, 53:205–216, 2017.

[155] Julia Ling, Andrew Kurzawski, and Jeremy Templeton. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *Journal of Fluid Mechanics*, 807:155–166, 2016.

[156] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. *arXiv preprint arXiv:1711.10561*, 2017.

[157] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics informed deep learning (part ii): Data-driven discovery of nonlinear partial differential equations. *arXiv preprint arXiv:1711.10566*, 2017.

[158] Julia Ling, Reese Jones, and Jeremy Templeton. Machine learning strategies for systems with invariance properties. *Journal of Computational Physics*, 318:22–35, 2016.

[159] Leon O Chua and Lin Yang. Cellular neural networks: Theory. *IEEE Transactions on circuits and systems*, 35(10):1257–1272, 1988.

[160] Yun Long, Xueyuan She, and Saibal Mukhopadhyay. Hybridnet: integrating model-based and data-driven learning to predict evolution of dynamical systems. *arXiv preprint arXiv:1806.07439*, 2018.

[161] Georg Martius and Christoph H Lampert. Extrapolation and learning equations. *arXiv preprint arXiv:1610.02995*, 2016.

[162] Subham S Sahoo, Christoph H Lampert, and Georg Martius. Learning equations for extrapolation and control. *arXiv preprint arXiv:1806.07259*, 2018.

[163] Zichao Long, Yiping Lu, and Bin Dong. Pde-net 2.0: Learning pdes from data with a numeric-symbolic hybrid deep network. *Journal of Computational Physics*, 399:108925, 2019.

[164] Kristina Monakhova, Joshua Yurtsever, Grace Kuo, Nick Antipa, Kyrollos Yanny, and Laura Waller. Learned reconstructions for practical mask-based lensless imaging. *Optics express*, 27(20):28075–28090, 2019.

[165] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

[166] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[167] Burhaneddin Yaman, Seyed Amir Hossein Hosseini, Steen Moeller, Jutta Ellermann, Kâmil Uğurbil, and Mehmet Akçakaya. Self-supervised physics-based deep learning mri reconstruction without fully-sampled data. *arXiv preprint arXiv:1910.09116*, 2019.

[168] Michael Kellman, Emrah Bostan, Nicole Repina, and Laura Waller. Physics-based learned design: Optimized coded-illumination for quantitative phase imaging. *IEEE Transactions on Computational Imaging*, 2019.

[169] Yunhao Ba, Guangyuan Zhao, and Achuta Kadambi. Blending diverse physical priors with neural networks. *arXiv preprint arXiv:1910.00201*, 2019.

[170] Russell Stewart and Stefano Ermon. Label-free supervision of neural networks with physics and domain knowledge. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[171] Chih-Wei Chang and NT Dinh. A study of physics-informed deep learning for system fluid dynamics closures. In *American Nuclear Society Winter Meeting, Anaheim, CA*, pages 1785–1788, 2016.

[172] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[173] Adrian Albert, Emanuele Strano, Jasleen Kaur, and Marta Gonzalez. Modeling global urbanization patterns with generative adversarial networks. In *IEEE International Geosciences and Remote Sensing Symposium*, 2018.

[174] Adrian Albert, Jasleen Kaur, Emanuele Strano, and Marta Gonzalez. Spatial sensitivity analysis for urban land use prediction with physics-constrained conditional generative adversarial networks. *arXiv preprint arXiv:1907.09543*, 2019.

[175] Luning Sun, Han Gao, Shaowu Pan, and Jian-Xun Wang. Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data. *arXiv preprint arXiv:1906.02382*, 2019.

[176] Olov Andersson, Fredrik Heintz, and Patrick Doherty. Model-based reinforcement learning in continuous environments using real-time constrained optimization. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[177] Jonathan Scholz, Martin Levihn, Charles Isbell, and David Wingate. A physics-based model prior for object-oriented mdps. In *International Conference on Machine Learning*, pages 1089–1097, 2014.

[178] Dehao Liu and Yan Wang. Multi-fidelity physics-constrained neural network and its application in materials modeling. *Journal of Mechanical Design*, 141(12), 2019.

[179] Nikhil Muralidhar, Mohammad Raihanul Islam, Manish Marwah, Anuj Karpatne, and Naren Ramakrishnan. Incorporating prior domain knowledge into deep neural networks. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 36–45. IEEE, 2018.

[180] Yinhao Zhu, Nicholas Zabaras, Phaedon-Stelios Koutsourelakis, and Paris Perdikaris. Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data. *Journal of Computational Physics*, 394:56–81, 2019.

[181] Nikhil Muralidhar, Jie Bu, Ze Cao, Long He, Naren Ramakrishnan, Danesh Tafti, and Anuj Karpatne. Physics-guided design and learning of neural networks for predicting drag force on particle suspensions in moving fluids. *arXiv preprint arXiv:1911.04240*, 2019.

[182] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016.

[183] Peter J Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics*, 656:5–28, 2010.

[184] Jonathan H Tu, Clarence W Rowley, Dirk M Luchtenburg, Steven L Brunton, and J Nathan Kutz. On dynamic mode decomposition: theory and applications. *arXiv preprint arXiv:1312.0041*, 2013.

[185] Bernard O Koopman. Hamiltonian systems and transformation in hilbert space. *Proceedings of the National Academy of Sciences of the United States of America*, 17(5):315, 1931.

[186] Igor Mezić. Analysis of fluid flows via spectral properties of the koopman operator. *Annual Review of Fluid Mechanics*, 45:357–378, 2013.

[187] Matthew O Williams, Ioannis G Kevrekidis, and Clarence W Rowley. A data–driven approximation of the koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6):1307–1346, 2015.

[188] Matthew O Williams, Clarence W Rowley, and Ioannis G Kevrekidis. A kernel-based approach to data-driven koopman spectral analysis. *arXiv preprint arXiv:1411.2260*, 2014.

[189] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Sparse identification of nonlinear dynamics with control (sindyc). *IFAC-PapersOnLine*, 49(18):710–715, 2016.

[190] Joshua L Proctor, Steven L Brunton, and J Nathan Kutz. Dynamic mode decomposition with control. *SIAM Journal on Applied Dynamical Systems*, 15(1):142–161, 2016.

[191] Joshua L Proctor, Steven L Brunton, and J Nathan Kutz. Generalizing koopman theory to allow for inputs and control. *SIAM Journal on Applied Dynamical Systems*, 17(1):909–930, 2018.

[192] Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *science*, 324(5923):81–85, 2009.

[193] Zichao Long, Yiping Lu, Xianzhong Ma, and Bin Dong. Pde-net: Learning pdes from data. *arXiv preprint arXiv:1710.09668*, 2017.

[194] Maziar Raissi. Deep hidden physics models: Deep learning of nonlinear partial differential equations. *The Journal of Machine Learning Research*, 19(1):932–955, 2018.

[195] Zhihua Wang, Stefano Rosa, Bo Yang, Sen Wang, Niki Trigoni, and Andrew Markham. 3d-physnet: Learning the intuitive physics of non-rigid object deformations. *arXiv preprint arXiv:1805.00328*, 2018.

[196] Guanya Shi, Xichen Shi, Michael O'Connell, Rose Yu, Kamyar Azizzadenesheli, Animashree Anandkumar, Yisong Yue, and Soon-Jo Chung. Neural lander: Stable drone landing control using learned dynamics. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9784–9790. IEEE, 2019.

[197] Yang Yu, Houpu Yao, and Yongming Liu. Physics-based learning for aircraft dynamics simulation. In *PHM Society Conference*, volume 10, 2018.

[198] Charles W Anderson. Learning to control an inverted pendulum using neural networks. *IEEE Control Systems Magazine*, 9(3):31–37, 1989.

[199] Sander Adam, Lucian Busoniu, and Robert Babuska. Experience replay for real-time reinforcement learning control. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(2):201–212, 2011.

[200] Burr Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009.

[201] Xiaojin Jerry Zhu. Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2005.

[202] Pinar Donmez, Guy Lebanon, and Krishnakumar Balasubramanian. Unsupervised supervised learning i: Estimating classification and regression errors without labels. *Journal of Machine Learning Research*, 11(Apr):1323–1351, 2010.

[203] Valeriy Gavrishchaka, Olga Senyukova, and Mark Koepke. Synergy of physics-based reasoning and machine learning in biomedical applications: towards unlimited deep learning with limited data. *Advances in Physics: X*, 4(1):1582361, 2019.

[204] Sunil Rao, Sameeksha Katoch, Pavan Turaga, Andreas Spanias, Cihan Tepedelenlioglu, Raja Ayyanar, Henry Braun, Jongmin Lee, U Shanthamallu, M Banavar, et al. A cyber-physical system approach for photovoltaic array monitoring and control. In *2017 8th International Conference on Information, Intelligence, Systems & Applications (IISA)*, pages 1–6. IEEE, 2017.

[205] Jiang Wan, Arquimedes Canedo, and Mohammad Abdullah Al Faruque. System level cps design using non-euclidean training method. 2018.

[206] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.

[207] Peng Li, Jens Eickmeyer, and Oliver Niggemann. Data driven condition monitoring of wind power plants using cluster analysis. In *2015 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*, pages 131–136. IEEE, 2015.

[208] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

[209] Mikael Henaff, Joan Bruna, and Yann LeCun. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163*, 2015.

[210] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, pages 3844–3852, 2016.

[211] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2008.

[212] Jonathan Masci, Davide Boscaini, Michael Bronstein, and Pierre Vandergheynst. Geodesic convolutional neural networks on riemannian manifolds. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 37–45, 2015.

[213] Václav Snášel, Jana Nowaková, Fatos Xhafa, and Leonard Barolli. Geometrical and topological approaches to big data. *Future Generation Computer Systems*, 67:286–296, 2017.

[214] Justin M Johnson and Taghi M Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1):27, 2019.

[215] Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intelligent data analysis*, 6(5):429–449, 2002.

[216] Nitesh V Chawla, Nathalie Japkowicz, and Aleksander Kotcz. Special issue on learning from imbalanced data sets. *ACM Sigkdd Explorations Newsletter*, 6(1):1–6, 2004.