A Review on Full-, Zero-, and Partial-Knowledge based Predictive Models for Industrial Applications

Information Fusion 119 (2025) 102996

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

A review on full-, zero-, and partial-knowledge based predictive models for industrial applications

Stefano Zampini [a,b], Guido Parodi [a,c], Luca Oneto [a,*], Andrea Coraddu [d], Davide Anguita [a]

[a] University of Genoa, Italy
[b] Polytechnic of Turin, Italy
[c] aizoOn Technology Consulting, Italy
[d] Delft University of Technology, The Netherlands

ARTICLE INFO

Keywords:
Predictive models
Industrial applications
Physical models
Data-driven models
Gray-Box models
Physics-informed models

ABSTRACT

In contemporary industrial applications, predictive models have been pivotal in bolstering production efficiency, product quality, scalability, and cost-effectiveness while promoting sustainability. These predictive models can be constructed solely based on domain-specific knowledge, exclusively on observational data, or by amalgamating both approaches. They are commonly referred to as Full-, Zero-, or Partial-knowledge-based predictive models, respectively. Full-knowledge based models are highly explainable, point-wise accurate, data efficient, computationally demanding in the prediction phase to achieve high accuracy. Zero-knowledge based models are poorly explainable, data hungry, highly accurate on average, computationally demanding in the construction phase but inexpensive the prediction phase. Partial-knowledge based models focus on taking the best of the two worlds. To maintain a focused scope and avoid redundancy with existing literature, our review will primarily delve into the key industrial applications within a narrow context, encompassing sectors such as extraction, chemical processes, manufacturing, transportation, energy, and construction. Our approach entails several steps. Initially, we conducted a meta-review to pinpoint gaps in previously published surveys on Full-, Zero-, and Partial-knowledge-based predictive models for industrial applications. Subsequently, we present a formal analysis of the subject matter, supplemented with illustrative examples to offer valuable insights. The core of our work comprises a review of existing research categorized by specific industrial applications. Finally, we outline the unresolved challenges and future prospects in this burgeoning field of research. We contend that our work serves as a valuable resource, catering to the needs of both young researchers seeking a solid foundation for their studies, industrial practitioners aiming to grasp core concepts and applications, and senior researchers seeking potential real-world applications for their findings.

Contents

* Corresponding author.
E-mail addresses: stefano.zampini@edu.unige.it (S. Zampini), guido.parodi@edu.unige.it (G. Parodi), luca.oneto@unige.it (L. Oneto), a.coraddu@tudelft.nl (A. Coraddu), davide.anguita@unige.it (D. Anguita).

https://doi.org/10.1016/j.inffus.2025.102996
Received 12 August 2024; Accepted 29 January 2025
Available online 10 February 2025

1. Introduction

The Industry has always been the ideal breeding ground for technological advancement and innovation to thrive [1–3]. From its birth till today (see Fig. 1), the industry went through a series of radical transformations that, in time, became more and more frequent [2]. The industrial transformation led also to changes in the evolution of society [2,4]. The 1st Industrial Revolution (late 18th to early 19th century) was facilitated by the introduction of the steam engine, the mechanized textile production, and iron manufacturing, which resulted in the mechanization of traditional industries, leading to increased productivity and urbanization [5]. The 2nd Industrial Revolution (late 19th to early 20th century) was propelled by the adoption of electricity, the internal combustion engine, and the expansion of the railroad network, resulting in the widespread use of electricity, the rise of the automobile industry, and the expansion of heavy industry [6]. The 3rd Industrial Revolution (Late 20th century) was driven by computers, digital technology, and the internet, leading to the digitization of information and automation of processes, transforming industries and communication [7]. The 4th Industrial Revolution (ongoing, began in the late 20th century) was facilitated by artificial intelligence, machine learning, blockchain, and the Internet of Things, characterized by the fusion of physical, digital, and biological technologies, revolutionizing industries, healthcare, and daily life [8,9]. Recent advances in artificial intelligence, quantum computing, advanced robotics, and nanotechnology raise the question of whether a 5th revolution is ongoing with a stronger focus on the societal impact of the tighter interaction between humans and machines [10].

In this work, we will use the term industrial applications, referring to sectors like extraction [11], chemical processes [12], manufacturing [13], transportation [14], energy [15], and construction [16]. Consequently, sectors like health [17], agriculture [18], utilities [19,20], and defense [21], which warrant separate comprehensive analyses, are intentionally excluded from this review.

In this context, the primary scopes of the industrial revolutions were to enhance many different Key Performance Indicators (KPIs) such as efficiency, flexibility, scalability, customization, automation, decentralization, quality [22]. While these KPIs have remained relatively consistent over time, while changing name and focus (e.g., from resource savings or waste minimization to sustainability), the tools for achieving and optimizing them have undergone significant changes [22]. In fact, the shift in the innovation tools driven by the most recent industrial revolutions represents a profound departure from traditional approaches [10,23,24]. While conventional innovation tools focus on incremental improvements to physical products, nowadays, the industry prioritizes disruptive innovation, integrating digital technologies, and data-driven decision-making [10,23,24]. This shift emphasizes customer-centricity and customization, enabling companies to tailor products to individual needs [24]. Additionally, modern industry embraces agility, rapid iteration, and ecosystem collaboration, fostering a dynamic and adaptable innovation landscape that can respond swiftly to changing market dynamics and customer preferences [3]. More in detail, digitalization means that products are increasingly connected and smart, incorporating sensors and data analytics to enhance functionality and provide real-time insights [3]. Mass customization is easier due to advanced manufacturing technologies allowing products to be tailored to individual customer needs [24]. Innovation now relies heavily on data analysis, enabling companies to gather valuable insights from user behavior and product usage, leading to continuous improvements [24]. Rapid prototyping, iterative design, and shorter development cycles are the norm, allowing for quicker adaptation to changing market demands [25]. Companies collaborate with partners and leverage open innovation platforms to access a broader pool of ideas and expertise [26].

The significance of the proposed review lies in its exploration of predictive models, which are central to the advancement of machine learning applications across various domains. These models, which include mathematical and computational algorithms, harness domain knowledge, computational resources, historical data, and statistical analysis to forecast future events or outcomes [27]. Often referred to as digital twins of real systems [28], predictive models serve as the cornerstone for the development of more sophisticated prescriptive models. Prescriptive models, which build on the predictive insights, not only anticipate future outcomes but also recommend optimal strategies to achieve specific objectives [29]. This survey will provide a comprehensive analysis of how these models are evolving in the context of the latest technological advancements, highlighting their critical role in the field and their potential to drive innovation in machine learning applications. In essence, predictive models show the path ahead, while prescriptive models guide organizations on how to navigate that path effectively [29] (see Fig. 2).

As a result, predictive models open up numerous opportunities for innovation [30]. Predictive models can help optimize industrial processes by forecasting outcomes and recommending actions [29]. This leads to more efficient resource utilization, reduced waste, and cost savings [31]. Predictive maintenance models can predict equipment failures before they occur, allowing for timely maintenance and preventing costly unplanned downtime [32]. Predictive models can identify defects and anomalies in real-time or in the production process, ensuring that products meet quality standards and reducing the need

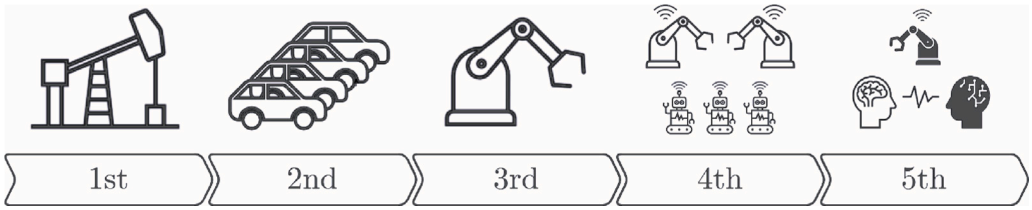

Fig. 1. Industrial revolutions.

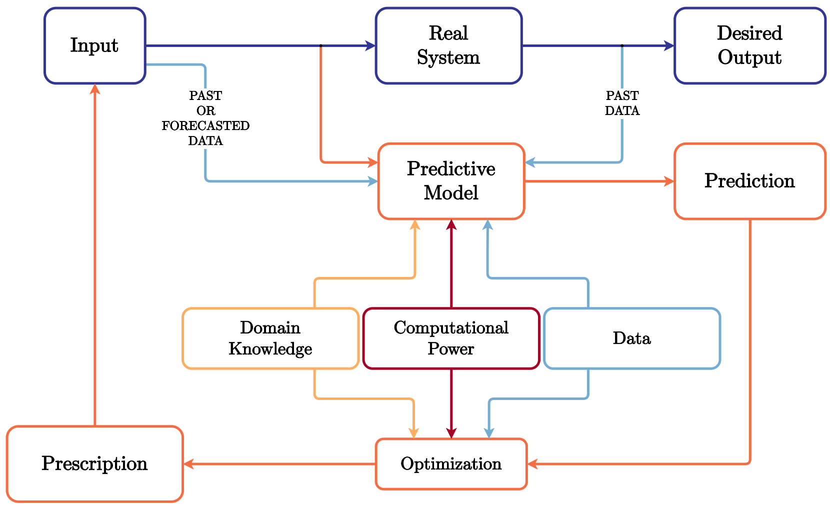

Fig. 2. Predictive models.

for post-production inspection and rework [33]. Predictive models can forecast demand for raw materials and finished products, helping companies optimize their inventory levels and reduce carrying costs while ensuring product availability [31]. Industrial processes often consume significant amounts of energy. Predictive models can optimize energy consumption by forecasting demand patterns and suggesting energy-saving strategies [30]. Predictive models can improve supply chain efficiency by predicting demand fluctuations, transportation delays, and optimizing logistics and distribution routes [34]. Predictive models can predict safety risks, such as equipment failures or hazardous conditions, allowing for preventive measures to protect workers and the environment [35]. In industries like chemical manufacturing, predictive models can optimize chemical reactions and process parameters to maximize yield and minimize waste [28]. Predictive models can analyze market trends and customer behavior to make informed decisions about production, pricing, and marketing strategies [30]. Companies that effectively leverage predictive models gain a competitive edge by making data-driven decisions, reducing costs, and improving product quality and customer satisfaction [27]. In highly regulated industries, predictive models can help ensure compliance with quality and safety standards [36]. Predictive models can allocate resources such as labor, machinery, and raw materials more effectively, improving overall operational efficiency [37]. Predictive models can assess and mitigate risks associated with various aspects of industrial operations, from supply chain disruptions to financial risks [31].

Predictive models can be constructed solely based on domain-specific knowledge [27,38–40] (i.e., full-knowledge based predictive models - FKPMs, or physical models, or White-Box models, or transparent models), exclusively on observational data [27,39,40] (i.e., zero-knowledge based predictive models - ZKPMs, or data-driven models, or Black-Box models, or machine learning based models), or by amalgamating both approaches [39,41,42] (i.e., partial-knowledge based predictive models - PKPMs, or hybrid models, or Gray-Box models, or physics-informed models).

FKPMs are built based on a-priori knowledge (e.g., physical laws or logical constraints) of the real system and can be very reliable and, by definition, explainable [39]. In fact, by construction, they only produce plausible predictions and can be easily inspected, corrected, and improved because of their full transparency [27]. They can also be empowered by using some historical data to estimate parameters of interest for the models [43]. The expected accuracy of the results improves with the increased amount of detail in modeling the physical phenomenon [44]. However, increasing the accuracy of FKPMs usually results in quite a high request in terms of computational requirements (e.g., computational fluid dynamic models) [44]. This fact limits their use in the wild, where huge computational capabilities are seldom available, making them unsuited for prescriptions [41].

ZKPMs, instead, do not require any a-priori knowledge about the system but rather are built on historical data (and some possible forecasted data like, e.g., weather conditions) collected from the real system [45]. ZKPMs are usually constructed using Machine Learning (ML) models. ZKPMs usually require a large amount of data and computational resources to be constructed (i.e., the learning phase) to reach satisfying performance in model accuracy [46]. Instead, once the model is constructed, its use for making predictions (i.e., the forward phase) is usually computationally inexpensive [47]. This has big added value for ZKPMs as only the forward phase needs to be exploited to use them in operation (e.g., for making prescriptions). For these reasons, these models are sometimes also leveraged to surrogate FKPMs to create a good approximation of FKPMs but with large computational advantages [40]. However, since they rely only on historical observations, ZKPMs work well statistically (i.e., on average) [47]. Still, they can produce implausible estimations (i.e., not physically plausible estimations) in particular situations [41]. Unfortunately, by nature, ZKPMs are Black-Box models that are hard to inspect (i.e., poorly explainable) and correct from these errors (which can also be due to spurious correlation or shortcuts learned by the model), making them unsuitable for safety-critical applications [48]. In particular, the major

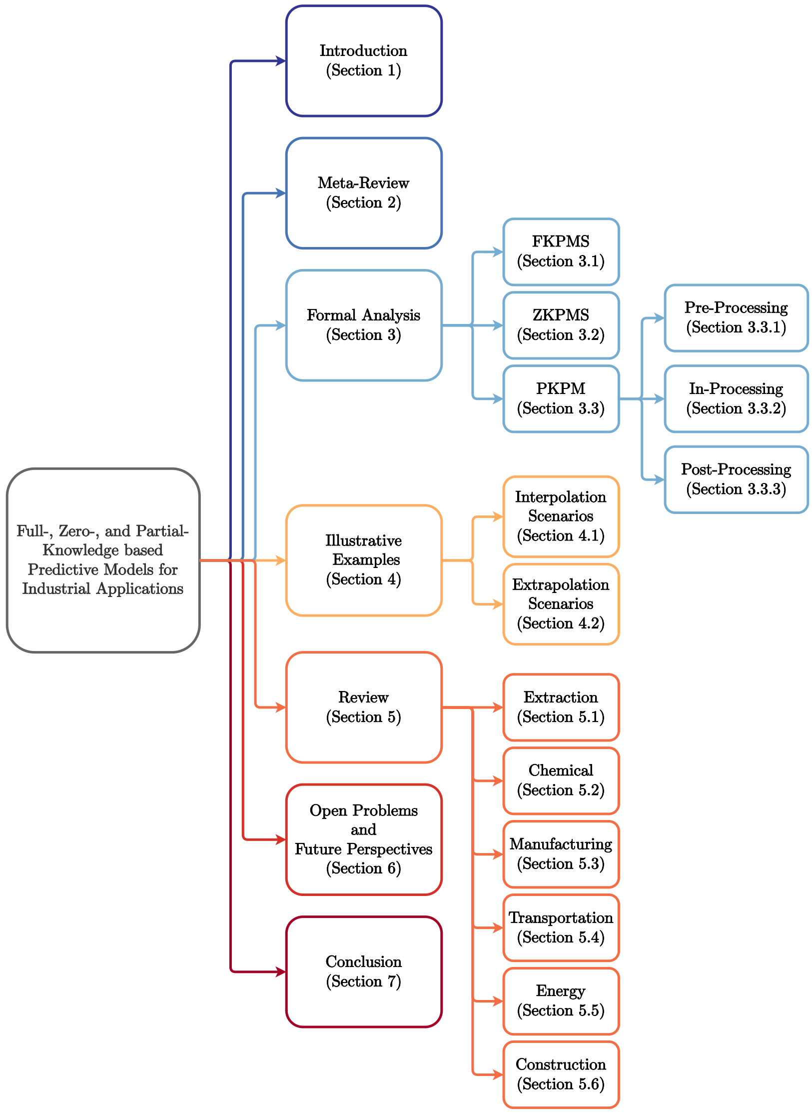

Fig. 3. Paper structure and organization.

lack of ZKPMs is that they commonly fail to make accurate predictions in extrapolating scenarios, namely in scenarios when ZKPMs are used in situations quite distant from the one observed in the data [39].

PKPMs have been recently introduced to fill both the gaps of FKPMs and ZKPMs and develop models able to take the best of the two worlds. PKPMs can exploit the a-priori knowledge about the real system and avoid implausible predictions, increase explainability, reduce the computational requirements of FKPMs by exploiting historical data, and reduce ZKPMs need for large amounts of historical data by starting from an already good approximation of the phenomenon provided by FKPMs. The idea of PKPMs is to start from a ZKPMs and to enrich it with domain knowledge in three different ways (that can also be combined together):

- pre-processing: the data are generated (e.g., when surrogating an FKPM [49,50]) processed (e.g., feature engineering [51], and data cleaning and cleansing [52]), or enriched (e.g., serial Gray-Box models [53], data augmentation [54]) leveraging the a-priori knowledge about the real system;
- in-processing: the ZKPMs leverage the a-priori knowledge during the learning phase (e.g., to decide the functional form of the model [55–57], to modify the training phase [58], the model selection or performance tuning phase [59]);
- post-processing: the prediction of the ZKPMs are "adjusted" leveraging the a-priori knowledge about the real system (e.g., parallel Gray-Box models [60], physical plausibility [61], and logic constraints [61]).

In this work, we argue that our review of FKPMs, ZKPMs, and PKPMs for industrial applications is able to

- conduct a meta-review to pinpoint gaps in previously published surveys on this topic underlying our contribution (see Section 2);
- provide a strong formal background of the subject (see Section 3);
- provide illustrative examples to offer valuable insights on the formal analysis (see Section 4);
- critically review the current work categorized by specific industrial applications (see Section 5);
- outline the unresolved challenges and future prospects in this burgeoning field of research (see Section 6).

Therefore, this review will serve as a valuable and fundamental resource, catering to the needs of young researchers seeking a solid foundation for their studies, industrial practitioners aiming to grasp core concepts and applications, and senior researchers seeking potential real-world applications for their findings. The organization of this review is also graphically depicted in Fig. 3.

2. Meta-review

In the literature, it is possible to find many different reviews partially dealing with the topics covered in our work. For this reason, this section reviews the current reviews to allow the reader to trace these works and better understand our novel contribution. In particular, Section 2.1 describes this review's search and selection criteria that will lead to the meta-review of Table 1, while Section 2.2 focuses on the gaps in the current reviews that will be addressed in our work.

2.1. Search and selection criteria

This section presents the search and selection criteria we adopted in our meta-review. In particular, we searched for reviews in a specific time window using a series of keywords in the academic database Scopus.[1]

For what concerns the time window, according to [35], Physics-informed ML, representing the latest wake of advancements in predictive models, began around 2016. The literature review on predictive models, until 2016, has been fully covered by the survey of [39]. For these reasons, our analysis will cover the last 10 years (i.e., from 2013) to cover also some precursor works that may have been overlooked [39].

For what concerns the keywords to be searched, we applied the following criteria

- Paper containing in the title, abstract, or keywords a series of strings referring to the context of this paper;
- Paper containing in the title, abstract, or keywords "review" or "overview" or "survey".

Since the results of this search[2] reported more than 500 papers (see in Fig. 4 the word cloud of their indexed Scopus keywords) we reduced this number filtering just the papers cited (according to Scopus)

- by ≥30 papers for works published from 2013 to 2021;
- by ≥20 papers for works published in 2022;
- by ≥10 papers for works published in 2023
- by ≥1 papers for works published in 2024.

Moreover, papers cited in relevant papers found in the application review (Section 5) have also been included.

The resulting papers have been filtered, selecting all the general or industry-related reviews. The result of this search is summarized in Table 1 where we reported: the work reference, the year of publication, the time span covered by the work, the analyzed FKPMs, ZKPMs, and PKPMs, the presence of a formal analysis, the presence of illustrative examples, and the domain of application. Fig. 5 shows the word cloud of their indexed Scopus keywords.

[1] https://www.scopus.com/

[2] This is the exact Scopus search string: TITLE-ABS-KEY(((theory-guided machine learning) OR (informed machine learning) OR (physics-informed machine learning) OR (physics-infused machine learning) OR (physics-guided machine learning) OR (physics-driven machine learning) OR (theory-guided data-driven) OR (physics-informed data-driven) OR (physics-infused data-driven) OR (physics-guided data-driven) OR (physics-based data-driven) OR (theory-guided neural network) OR (theory-guided neural networks) OR (physics-informed neural network) OR (physics-informed neural networks) OR (physics-infused neural network) OR (physics-infused neural networks) OR (physics-guided neural network) OR (physics-guided neural networks) OR (physics-driven neural network) OR (physics-driven neural networks) OR (gr?y-box) OR (hybrid data-driven) OR (hybrid modeling) OR (hybrid modelling) OR (hybrid machine learning) OR (hybrid data driven)) AND (review OR overview OR survey)) AND PUBYEAR > 2022.

2.2. Findings

From Table 1, we can derive a series of observations regarding the current reviews on the topics faced in this work and our contribution.

Although they have different nomenclatures, all the reviews agree that predictive models should be classified into FKPMs, ZKPMs, and PKPMs.

Every review describes FKPMs as methods that infer relationships between physical variables, according to physics [27]. According to [39], FKPMs can be Physics-Equation-based (e.g., Algebraic equations, Ordinary differential equations -ODE-, or partial differential equations -PDE-) or State-Machines-based (Mealy or Moore machines). A general formal analysis of FKPMs is present in [39], while [27,62,66], and [82] provide domain-specific ones.

The different definitions of ZKPMs can be summarized as "models that completely rely on data" [39]. ZKPMs include all the techniques under the Supervised Machine Learning umbrella, e.g., Shallow [84] and Deep Models [84], Frequentist [85] and Bayesian models [86], and Binary or Multiclass Classification [87] and Regression [88]. In [39], a brief analysis of ZKPMs models is presented, while in [27,62,66], and [78] an analysis of their application to different domains is provided.

Regarding PKPMs, every review agrees with a simple definition, "models leveraging both on domain knowledge and available data", but a comprehensive analysis of the methods and application domains is not reported. In literature, it is possible to find precursor surveys like [62,63], reviewing hybrid methods (e.g., serial and parallel PKPMs) that are a sub-family of PKPMs that simply combine FKPMs and ZKPMs, without modifications of their inner structure, that have been used since the 1990s. In fact, as described in [66,89] was the very first work proposing to leverage on a-priori knowledge for the neural networks implementation, while [90] presented the first application of a serial combination of a ZKPM and an FKPM. As per [63], the first theoretical analysis of serial and parallel combination was addressed by [91]. Over time, the integration of FKPMs and ZKPMs has become increasingly tight. According to [35,42] first introduced "Theory-guided data science" as integrating scientific theories into data science. Many surveys [35,41,72,74,79,81,92], recognize the works of [58,93,94] as a breakthrough in the field of PKPMs. The authors proposed the integration of differential equations (i.e., Physics-Equation-based FKPMs) and neural networks (i.e., specific ZKPMs) by modifying the loss function leveraged during the training phase of the neural networks. The first review of PKPMs was proposed by [72], defining a PKPM as the combination of an FKPM (i.e., a domain knowledge source) and a technique of integrating it inside the ZKPM (i.e., a data-driven model). The work of [41] is currently the most popular review about PKPMs, in which the authors presented three ways of informing a ZKPM employing an FKPM (i.e., observational, learning, and inductive biases). Other notable works are the one of [39], which approaches the development of Cyber Physical Systems with FKPMs, ZKPMs, and PKPMs, and [27], which analyzes recent trends of hybrid modeling for smart manufacturing.

Based on this meta-review it is possible to find the following gaps

- a formal analysis of FKPMs, ZKPMs, and PKPMs able to cover and generalize the three families of PMs with a uniform and consistent taxonomy for PKPMs is still missing in the literature since each survey introduces a different classification method partially covering the FKPMs, ZKPMs, and PKPMs landscapes;
- simple illustrative example to understand the advantages and disadvantages of FKPMs, ZKPMs, and PKPMs is still missing;
- a review covering all ways in which FKPMs, ZKPMs, and PKPMs have been leveraged in Industrial Application (in a sense explained in the introduction) is still missing since most reviews focused on a very narrow domain.

For this reason, this review will focus on filling these four gaps in Sections 3, 4, and 5, respectively.

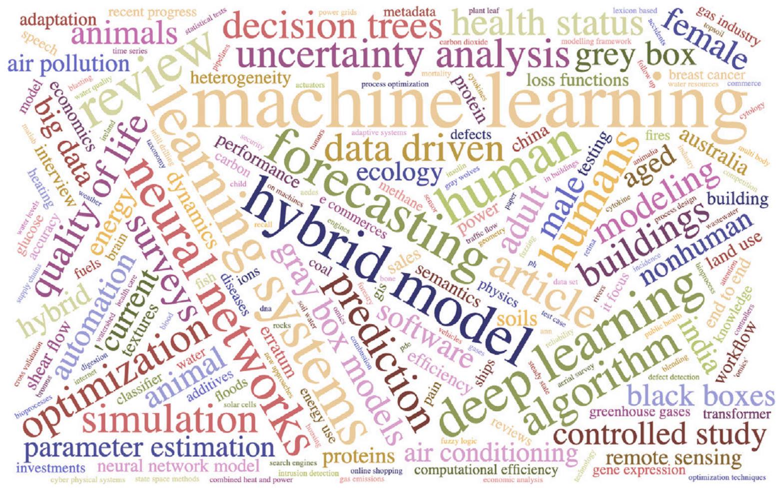

Fig. 4. Word cloud of the indexed Scopus keywords of all the retrieved works from Scopus for the meta-review.

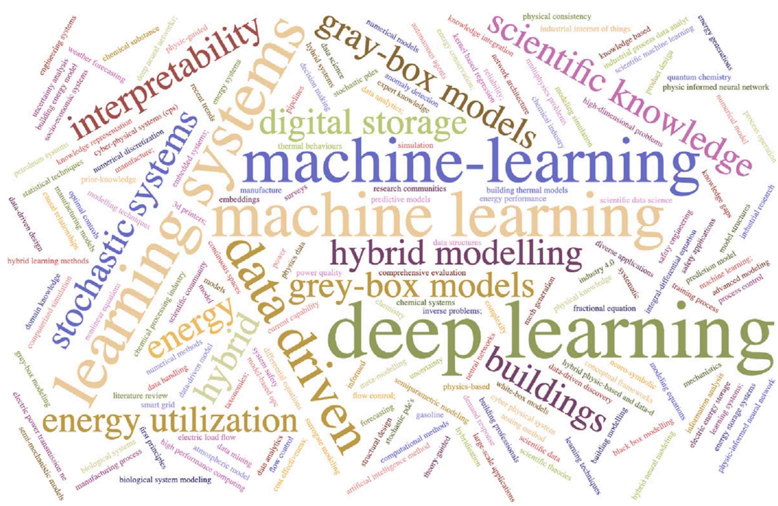

Fig. 5. Word cloud the indexed Scopus keywords of the works reported in Table 1.

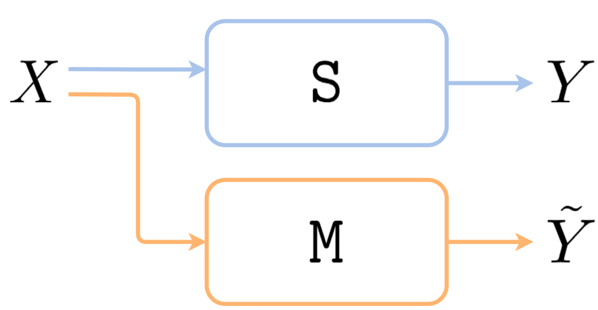

Fig. 6. Predictive model.

3. Formal analysis

In this work, we deal with the problem of building predictive models. We have a system S that, given some input $X \in \mathcal{X}$, where $\mathcal{X}$ is the input space, produces some output $Y \in \mathcal{Y}$, where $\mathcal{Y}$ is the output space. The scope is to build a model M of S that, given only X can produce $\tilde{Y} \in \mathcal{Y}$ such that $Y \approx \tilde{Y}$ according to a desired metric $\mathscr{M}$. This concept is graphically depicted in Fig. 6.

Note that S may be both deterministic and non-deterministic [95], and $\mathcal{X}$ and $\mathcal{Y}$ may be, e.g., a d-dimensional real vector [95], a graph [96], a tree [97], an image [98], natural language [99], or a mix of the aforementioned information. Note also that, in general, there may not be any causal link [100] between X and Y but just correlation [95,101]. Finally, note that X usually contains things that are easily or cheaply measurable while Y usually contains things that are hard or costly to measure and, consequently, may produce value [102].

This section will present a formal analysis of the M construction using FKPMs, ZKPMs, and PKPMs. M is usually defined as a model f chosen in a set of possible ones $\mathcal{F}$. In this work, formal analysis refers to systematically examining the model's mathematical, statistical, or logical structure to understand its behavior under various conditions. This process is instrumental in identifying the underlying assumptions, limitations, and potential biases inherent in the models facilitating the identification of optimal model parameters and structures, and contributing to the development of more efficient, effective, and reliable predictive tools.

In order to design and build M we may have different sources of information. We report here the four main ones.

- **Domain Knowledge**: this source of information refers to the pre-existing understanding of the dynamics and operational principles governing S. Such knowledge can manifest in various forms. One example is the physical laws that are comprehensive or partial descriptions of the causal relationships between inputs and outputs within the system [41]. A second example may be

Table 1
Related works in chronological order..

Review	Year	Timespan	FKPMs	ZKPMs	PKPMs	Formal Analysis	Practical Example	Domain
[62]	2013	Unspecified	Computational Fluid Dynamics, Zonal approach, Nodal approach	Multiple Linear Regression, Genetic Algorithm, Neural network, Support vector machine	Hybrid models	Yes	No	Buildings' Energy
[63]	2014	1992–2013	Linear/nonlinear Ordinary/Partial differential equations, Linear/Nonlinear algebraic equations	Neural networks, Partial least squares, Splines, Kernels	Serial, Parallel	Partial (PKPMs)	No	Chemical, Biochemical
[64]	2014	Unspecified	White-Box	Black-Box	Gray-Box	No	No	Buildings' Energy
[65]	2016	2000–2016	White-Box	Black-Box	Gray-Box, Hybrid models	No	No	Vehicle Fuel Consumption
[42]	2017	Unspecified	No	No	Theory-guided design, Theory-guided learning, Theory-guided refinement, Learning hybrid models of theory and data-science, Augmenting theory-based models	Partial (PKPMs)	No	Science
[66]	2018	Unspecified	White-Box	Black-Box	Gray-Box (Parallel, Serial, Mixed parallel/serial)	Yes	No	Chemical, Petroleum, Energy
[67]	2018	2007–2017	White-Box	Black-Box	Gray-Box	No	No	Buildings' Energy
[39]	2020	Unspecified	Physics equation based CPS, State Machine Based Cyber-Physical System	Data-driven CPS	Physics-based Pre-processing, Physics-based Network Architectures, Physics-based Regularization, Miscellaneous	Yes	No	General
[68]	2020	Unspecified	Simulation	Machine Learning	Simulation-assisted ML, ML assisted simulation	No	No	General
[28]	2020	Unspecified	No	No	Serial, Parallel	No	No	Chemical
[69]	2020	Unspecified	White-Box	Black-Box (Neural networks, Support vector machine, Random forest, Gradient boosting, Multiple linear regression)	Gray-Box	No	No	Buildings' Energy
[70]	2020	Unspecified	No	No	Physics-guided loss function, Physics-guided initialization, Physics-guided design of architecture, Residual modeling, Hybrid physics-ML models	No	No	General
[71]	2021	Unspecified	No	No	Gray-Box	Yes	No	Buildings' Energy

(continued on next page)

the input/output insensitivity, which pertains to the invariance of the output against variations in the input under specific conditions [103]. A final example is the output constraints, which include any prerequisites or limitations imposed on the output, ensuring it adheres to certain criteria or standards [104]. Domain knowledge plays a crucial role in understanding and predicting the behavior of $\mathbb{S}$, offering valuable insights into its functioning and facilitating the development of more accurate and efficient predictive models.

- **Measures**: this source of information is intrinsically linked to domain knowledge as it often enables the direct measurement or collection of data pertaining to the characteristics of a system [102]. For instance, it is feasible to ascertain specific attributes, such as machinery's actual weight, or to gather data, such as its plate information. Such empirical data are invaluable, particularly in the context of developing FKPMs (see Section 3.1). These models necessitate precise information about $\mathbb{S}$ to be effectively constructed and validated.

Table 1 (*continued*).

[40]	2021	1992–2020	Mechanistic models	Data-driven models (Neural networks, Support vector machine, Multivariate Adaptive Regression Splines, Latent Variable Models)	Serial, Parallel, Surrogate, Alternative models	No	No	Chemical
[72]	2021	Unspecified	No	No	Knowledge source, Knowledge representation, Knowledge integration's way (training data, hypothesis set, learning algorithm, final hypothesis)	Partial (PKPMs)	No	General
[41]	2021	Unspecified	No	No	Observational bias, Inductive bias, Learning bias	Partial (PKPMs)	Yes	General
[73]	2021	Unspecified	White-Box	Black-Box	Gray-Box	No	No	Batteries
[74]	2022	Unspecified	No	No	Mechanistic feature processing, Physics-informed model development, Data-driven discovery	Partial (PKPMs)	No	Manufacturing
[75]	2022	2014–2021	No	No	Data, cost function, initialization, run time, architecture	No	No	Civil Engineering
[76]	2022	Unspecified	No	No	ML compliments science (Inverse models, direct hybrid models, reduced order models, uncertainty quantification, discover laws), Science compliments ML (design, learning, refinement)	No	No	Chemical
[77]	2022	Unspecified	No	No	Physics-informed loss function, Physics-informed initialization, Physics-informed design architecture, Hybrid physics-DL models	Partial (PKPMs)	No	Power Systems
[27]	2022	Unspecified	Physics-based models	Data-driven models	Physics-informed ML (data, modeling, loss function), ML assisted simulation, Explainable Artificial Intelligence	Yes	No	Manufacturing
[78]	2022	Unspecified	No	Gaussian Process, Neural networks	Hybrid sub-modeling, Physics-informed ML, Model calibration	Partial (ZKPMs, PKPMs)	Yes	General
[79]	2022	2018–2022	No	No	Physics-informed neural networks	Partial (PKPMs)	No	Science
[80]	2022	2001–2021	White-Box	Black-Box (Statistical and ML models)	Gray-Box (Black-Box based on White-Box and White-Box based on Black-Box)	No	No	Ship Fuel Consumption
[81]	2022	Unspecified	No	No	Physics-informed data, architecture, loss function, optimization, inference	Partial (PKPMs)	No	General

(*continued on next page*)

- **Example of the Input/Output relation (Dataset)**: this source of information is typically required by the ZKPMs (see Section 3.2). In the era of big data, an extensive array of information has been amassed and archived across various systems [105]. Although collecting input data is generally cost-effective, acquiring output data can be challenging due to its inherent value [102]. The collection of output data is frequently expensive or, in some instances, unfeasible if the outputs are only determinable in the future [102]. However, within historical datasets, such output information might be readily and economically accessible. Formally, a dataset [95] is characterized as a collection of n instances demonstrating the input/output relationship, denoted as

Table 1 (*continued*).

[82]	2022	2002–2022	Physics-based (Electrochemical, Equivalent circuit, Semi-empirical)	No	Gray-Box (Data-driven assisted physical models, Physics-guided data-driven)	Partial (FKPMs, PKPMs)	No	Batteries
[83]	2022	Unspecified	No	No	Physics-guided loss function, Physics-guided initialization, Physics-guided design of architecture, Hybrid physics-ML models	Partial (PKPMs)	No	Engineering, Environmental Systems
[35]	2023	2016–2022	No	No	Physics-informed loss function, Physics-informed architecture	Partial (PKPMs)	No	General

$\mathcal{D}_n = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$. In certain cases, particularly historical ones, obtaining output data for every input may remain prohibitively expensive or impossible. This scenario necessitates the compilation of an alternative dataset, referred to as unsupervised [106], represented as $\mathcal{U}_m = \{X_1, \dots, X_m\}$. Despite this dataset's absence of output data, it can still offer significant value for specific applications [107,108].

- **Experience**: this source of information is the most complex one to describe since it cannot be formalized as the previous ones. For example, in FKPMs (see Section 3.1), experience allows us to disregard some effects since previous analyses on the same topic resulted in this know-how. In ZKPMs (see Section 3.2), experience may lead to choosing a particular ML algorithm rather than another one or to choosing a particular deep learning architecture. Finally, in PKPMs (see Section 3.3), experience may lead to choosing the most suited way to blend FKPMs and ZKPMs. In synthesis, the experience can be defined as all how users and developers of FKPMs, ZKPMs, and PKPMs reduce the number of free choices, according to their previous knowledge, experience, or know-how [72]. This, in practice, can make more of a difference with respect to actual technical skills [72].

These sources can be present simultaneously or not, and the availability of specific sources determines whether one can construct FKPMs, ZKPMs, or PKPMs

Another important aspect to consider is the timing associated with both X and Y during the model's construction and usage. In fact, model construction is a retrospective work where we can assume to know everything [47] (both past, present, and future). During model operation, instead of just knowing some past and present information, we may have an estimation of some data about the future [109] (e.g., weather forecast), and we usually want to predict some present information [110] (i.e., nowcasting) of more probably, some future information [111] (i.e., forecasting). Note that present and future may not be simply associated with the actual flux of time but they are associated with the actual time in which that information is available. For instance, we may know now if a machinery component is worn, but we will know it just when this problem generates a fault or when an operator checks and measures it. In this perspective, predicting something about the past (i.e., backcasting [112]) can still be useful if that information is lost (and we need that for, e.g., diagnostics purposes) or that past condition will be labeled in the future.

To provide a clearer understanding of this concept, we illustrate it with the graphical representation shown in Fig. 7.

The remaining part of this section is devoted to the formal analysis of how FKPMs (Section 3.1), ZKPMs (Section 3.2), and PKPMs (Section 3.3) can be designed and operated. In other words, will describe what happens in the red dashed blocks of Fig. 8.

3.1. Full knowledge predictive models

FKPMs, in general, require all four sources of information to be built [38]. They are the oldest way to build predictive models and, historically, the ones that have been more used until the advent of the digital era [40]. As with all $\mathtt{M}$ they are characterized by a model f chosen in a set of possible ones $\mathcal{F}$ and they can be designed as follows (see Fig. 9)

- the predictive model f takes as inputs a subset of the possible ones $X' \subseteq X$ since it may be too complex or not possible, based on the domain knowledge, to model the relation between Y and some X. Moreover, based on the experience, it may be worthy or worthless to actually model some relation since it is important or negligible for the specific application;
- the functional form of f (e.g., linear or differential equations) depends on the domain knowledge. For some systems $\mathtt{S}$, we may have just simplified relations to model a phenomenon (e.g., empirical formulas), while for others, we may have the exact solution (e.g., exact differential equations). The functional form of f can also be generated incrementally by many interconnected blocks, which adds hierarchy and interpretability to the model;
- the parameters θ that characterize f can be subdivided in three main groups θ_K, θ_M, and θ_D such that $\theta = \theta_K \cup \theta_M \cup \theta_D$, $\theta_K \cap \theta_M = \oslash$, $\theta_K \cap \theta_D = \oslash$, and $\theta_M \cap \theta_D = \oslash$, and
  - θ_K are set based on the domain knowledge (e.g., the universal constant of gravitation);
  - θ_M are set based on the measures (e.g., the mass of a machinery);
  - θ_D are tuned by an algorithm $\mathscr{A}_H$, characterized by its hyperparameters H that leverages: the available $\mathcal{D}_n$ (to guide the selection of the optimal parameters), the experience (to guide the selection of the algorithm and its hyperparameters) and the domain knowledge (to give a range of possible values of θ_D, namely $\mathcal{F}$).

The definition/creation of the functional form f of an FKPM is something that may take hours, days, or even years to be formulated since it is heavily guided by humans [68].

The definition/creation of the functional form f of an FKPM is often deeply dependent and crafted toward a particular application [38]. For example, a particular phenomenon in certain operating conditions can be well approximated by a simple empirical formula, while in other operating conditions, the only good solution is the one of solving a complex set of partial differential equations [38].

The search of θ_D can be performed with a grid search when the number of parameters is very low (i.e., less than 10) otherwise, we need to formulate the problem as an optimization problem that can be convex (then easy to solve) or non-convex by differentiable (less easy but still manageable with multi-start gradient descent algorithms) or non-convex (then manageable just employing meta-heuristic algorithms or with differentiable surrogates) [113]. Each optimizer has its hyperparameters to be tuned, and, in this case, only experience coupled with grid search can help us in the process [114]. In all cases, the best main idea behind the search for the best values of θ_D, namely θ_D^* can be formulated as follows

$$\theta_D^* = \arg\min_{\theta_D \in S} \mathrm{E}(\theta_D, \mathcal{D}_n), \tag{1}$$

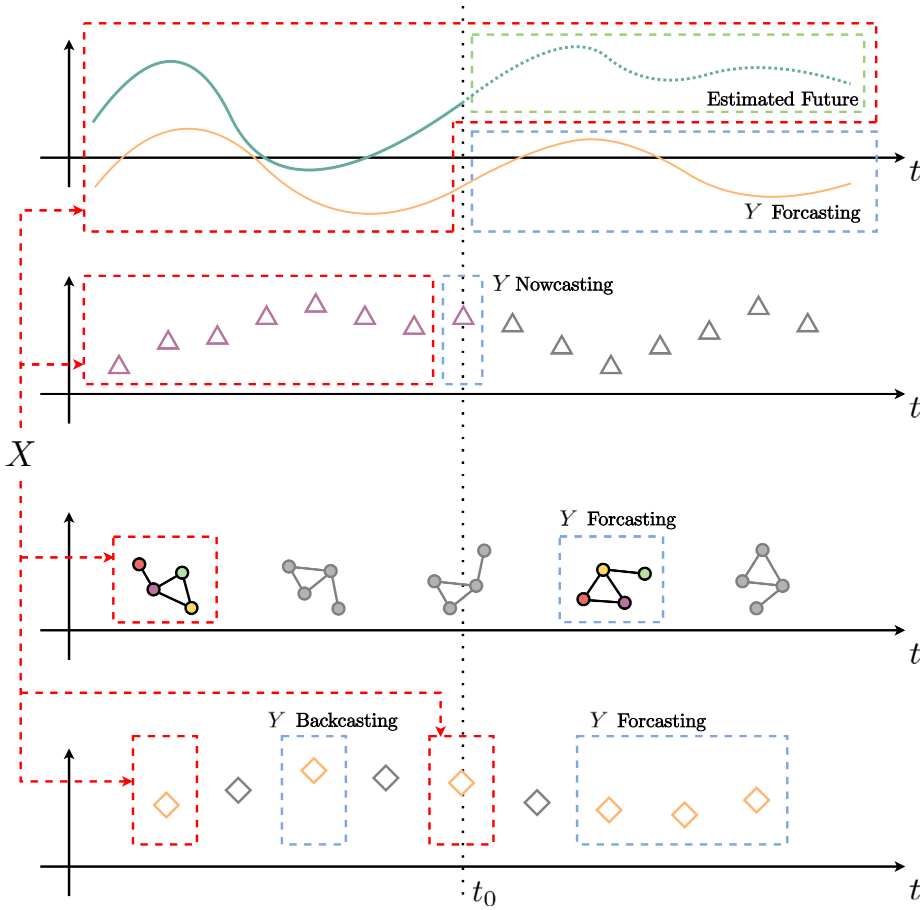

Fig. 7. Past, present, and future in designing M of Fig. 6. Different quantities that vary in time (continuous/discrete structured/unstructured quantities). At the time t_0 (present), we know some of them (dashed red boxes) because they are part of the estimated future information used as input. As output (making some back-, now-, fore-casting), there is some quantity of interest that we do not know (dashed blue boxes).

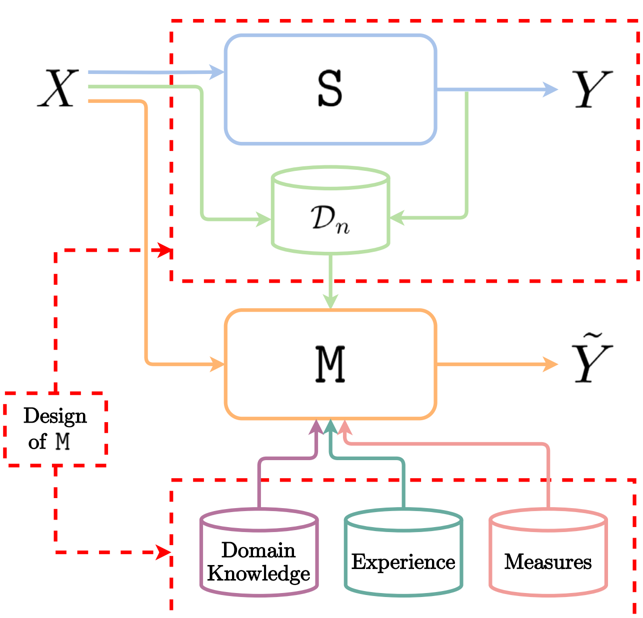

Fig. 8. Design M of Fig. 6.

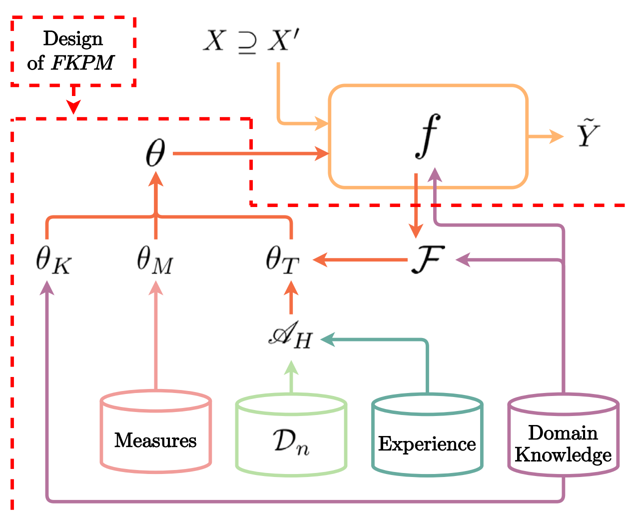

Fig. 9. FKPMs.

where $\mathcal{S}$ is the search space (induced by domain knowledge or experience) and $\mathrm{E}(\theta, \mathcal{D}_n)$ is the error, measured with reference to the desired metric [115], determined by the specific θ_D over the data $\mathcal{D}_n$ and $\mathscr{A}_H$ is the algorithm, together with it hyperparameters H, leveraged to solve the min problem.

Depending on the application, the needed accuracy, the input available, how much S has been studied in the past, and the computational power, one may find many different solutions toward better FKPMs [38].

FKPMs have different advantages and disadvantages. Regarding the advantages FKPMs

- they require a relatively small amount of data to be constructed [40];
- they work well in interpolation (i.e., close to the available measure or the design condition) [116];
- they work reasonably well in extrapolation (i.e., far away from the available measure or the design condition). By reasonably well, we meant that maybe they are not perfectly accurate, but they always provide behavior that is physically plausible by construction [116];
- they are explainable and understandable by construction. As a consequence, they can be easily inspected, tested, and queried and they can provide new insight into the problem [117];
- in some cases they require a small amount of computational power when used to make predictions (e.g., empirical models) [62].

Regarding the disadvantages FKPMs

- they require a lot of human intervention and time to be designed, constructed, and tested [38];
- they do not fully exploit all the measures (inputs) that nowadays can be retrieved in a real application [43];
- in some cases their accuracy is low due to their inability to model complex phenomena or to leverage all the data [62];
- in some cases, they require a huge amount of computational power when used to make predictions (e.g., numerical solution of complex partial differential equations), preventing their use for design optimization purposes [44].

3.2. Zero knowledge predictive models

ZKPMs, in general, require only two sources of information to be built: the dataset and experience [47]. They are application-independent and do not need any domain knowledge about S or measures about the systems (i.e., constants that characterize the phenomena) [47]. Thanks to the availability of large amounts of data and computational power, they became popular in the last twenty years, the digital era [40]. Also in this case, the M are characterized by a model f chosen in a set of possible ones $\mathcal{F}$ and they can be designed as follows (see Fig. 10)

- the predictive model f takes as inputs all the available ones X since that can handle a huge number (the only limit is the computational power) of different input sources (e.g., scalars, vectors, time series, images, graphs, and natural language). Note that ZKPMs do not distinguish between causality and correlations, so it may be risky to put "too much" into X since doing so increases the probability of stumbling upon a spurious correlation. This concept is referred to in ZKPMs with the simplified sentence "garbage in, garbage out";
- the functional form of f depends on two main types of parameters. The hyperparameters that regulate the functional form in this field, called architecture, of the model (e.g., linear, nonlinear, kerneled, convolutive, attention, and recursive) while the parameters are the ones that fixed the hyperparameters, define a specific model $\mathcal{M}$. Hyperparameters inherit their name from the fact that they strongly influence the performance of the model both positively (as for FKPMs, sometimes more complex models lead to higher performance) and negatively (too much complexity may lead to poor performance; see later the problem of overfitting). The functional form of f is usually monolithic, i.e., given and X produces directly the Y, with no intermediate blocks that have specific hierarchy or interpretability properties;
- since we do not have any domain knowledge that guides our choice, the only information about the quality of our model M must be extracted from $\mathcal{D}_n$. This leads to an immediate problem: we cannot use the same data to build, optimize, and estimate the model's performance; otherwise, we risk obtaining optimistic results. This problem, in ZKPMs, is called data-snooping [118], which leads to problems commonly referred to as over-fitting, over-validation, and over-test. The standard solution is to assume that the dataset in $\mathcal{D}_n$ is independent and identically distributed (i.e., i.i.d.), split (one or multiple, r, times) the data in three sets called learning $\mathcal{L}_l^i$, validation $\mathcal{V}_v^i$, and test $\mathcal{T}_t^i$ such that $\mathcal{D}_n = \mathcal{L}_l^i \cup \mathcal{V}_v^i \cup \mathcal{T}_t^i$ and $\mathcal{L}_l^i \cap \mathcal{V}_v^i = \emptyset$, $\mathcal{L}_l^i \cap \mathcal{T}_t^i = \emptyset$, and $\mathcal{V}_v^i \cap \mathcal{T}_t^i = \emptyset$ with $i \in \{1, \dots, r\}$ then
  - use $\mathcal{L}_l$ to optimize the parameters θ that characterize f using an algorithm $\mathscr{A}_H$, characterized by its hyperparameters H, that tries to find the θ that minimizes the error, measured with some metric, of f on $\mathcal{L}_l$ subject to some constraint on θ. In other words, $\mathscr{A}_H$ chooses $f \in \mathcal{F}$, where $\mathcal{F}$ is more or less explicitly induced by the constraint on θ, $\mathscr{A}$, or H. These problems are usually convex (allowing finding the optimal point efficiently) or at least differentiable (allowing finding a local minima efficiently via gradient descent). Note that always exists a combination of $\mathscr{A}$ and H that leads to an arbitrarily small error on $\mathcal{L}_l$, which is, in general, not good since we do not want to simply memorize the information in $\mathcal{L}_l$ (over-fitting), but we want to work well on previously unseen data (generalize). In other words we need to carefully tune $\mathscr{A}$ and associated H;
  - then we need to use $\mathcal{V}_v$ to optimize the choice of the algorithms $\mathscr{A}$ and its hyperparameters H finding $\mathscr{A}$ and associated H that minimizes the error, measured with some metric, of the f generated by $\mathscr{A}_H$ on $\mathcal{V}_v$. In other words, f generated by $\mathscr{A}_H$ using $\mathcal{D}_n$ should work well on the previously unseen (and i.i.d.) data $\mathcal{V}_v$; These problems are

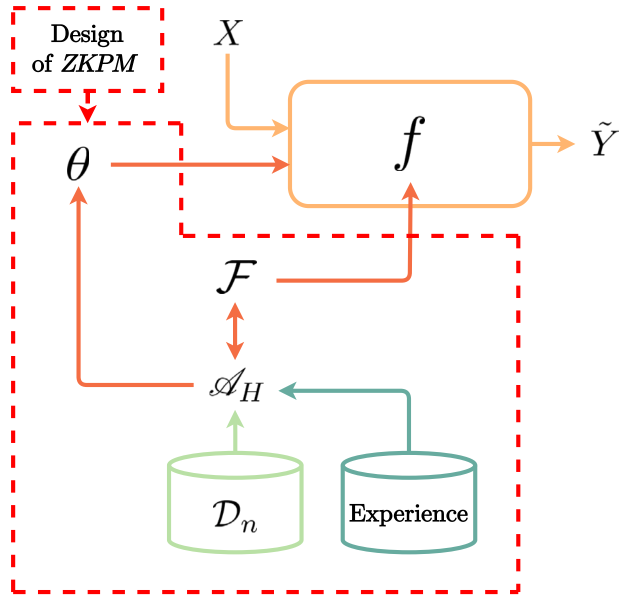

Fig. 10. ZKPMs.

usually non-convex and not-differentiable globally. For this purpose, it is commonly addressed with some global grid search (e.g., search for different algorithms) in combination with some local optimization of differentiable or differentiable surrogate problems (e.g., search for the optimal value of a hyperparameter). Note that experience may lead to some pruning to the choices to check against the data. In this case, if the number of choices for $\mathscr{A}$ and relative H is too high, we risk making a choice that is only able to minimize the error on $\mathcal{V}_v$ and not on previously unseen data. So we cannot trust the error that the model f generated by the best $\mathscr{A}$ and relative H performs on $\mathcal{V}_v$;
– then we need to use $\mathcal{T}_t$, a fresh new (and i.i.d.) set of data with respect to $\mathcal{L}_l$ and $\mathcal{V}_v$ to check the performance of f generated by the best $\mathscr{A}$ and relative H using $\mathcal{L}_l \cup \mathcal{V}_v$. This error, by definition, is a statistically unbiased estimator of the error that the f generated by the best $\mathscr{A}$ and relative H will perform on previously unseen data since $\mathcal{T}_t$ is only used once. A common mistake [119,120] is to reuse $\mathcal{T}_t$ multiple times, performing the so-called over-test or data-snooping [118], which is a big mistake in practical situations: from the evaluation procedure we expected a certain level of accuracy and then, when the model is actually used in production, the accuracy is much less.

This approach, which is at the basis of all resampling approaches, is commonly used by practitioners, but other, more theoretical approaches exist [121].

One problem of the i.i.d. assumption is that it is never verified in practice [47]. The solution to this problem is to make the datasets $\mathcal{L}_l$, $\mathcal{V}_v$, and $\mathcal{T}_t$ verify the i.i.d. assumption as much as possible [122]. Let us give some examples. If we need to recognize images, we need to be sure that $\mathcal{D}_n$ contains images coming from different places (e.g., varying background) and time (e.g., varying light exposition). This problem is called the design of experiments for data collection [123]. Another problem is when we want to predict a time series. In this case, we cannot put random samples (past and future samples) in $\mathcal{L}_l$, $\mathcal{V}_v$, and $\mathcal{T}_t$ but we always need to keep the sequence in time, namely $\mathcal{L}_l$, $\mathcal{V}_v$, and $\mathcal{T}_t$ should contain data from $[t_1, t_2)$, $[t_2, t_3)$, and $[t_3, t_4]$ respectively with $t_1 < t_2 < t_3 < t_4$ [124].

Note also that the algorithm $\mathscr{A}$, usually called supervised ML algorithms in this context, are characterized by its hyperparameters H, selects a ZKPMs $(\mathtt{M})$ $f \in \mathcal{F}$ based on the available data such $f = \mathscr{A}_H(\mathcal{D}_n)$ and many different ML algorithms exist in the literature [95,125]. One may think that, at least theoretically, there should exist at least an optimal $\mathscr{A}$ simplifying the selection of it. Instead, the no-free-lunch theorem [126] ensures that there is no way to determine a-priori the best ML algorithms to use for a specific application.

Many different ML algorithms and families of algorithms exist in the literature. Nevertheless, the most commonly and effective families[3] are

- kernel methods [127];
- ensemble methods [128];
- neural network [98];
- Bayesian approaches [129];

some of them also empowered by fuzzy techniques [130,131]. More abstractly, there are two main approaches to ML. When data is tabular (or more properly unstructured), usually shallow ML models are employed [95]. Shallow models directly manipulate X to generate Y, which is considered a good representation for making effective predictions. When data is structured (e.g., images, graphs, and natural language), usually, deep ML models are employed [98] . Deep models are exploited since directly manipulating X to generate Y is not effective (and not possible in some cases), and it is needed to first learn a good representation of the input data (in fact, deep learning is sometimes referred to as representation learning) from the data and then, using this representation, effective predictions are generated. This also opened the way to procedures like transfer learning [132,133] with pre-trained[4] and foundation[5] models (i.e., reuse the same representation for multiple tasks).

In order to tune the parameters θ of f, all ML models mostly rely on the Empirical Risk Minimization (ERM) principle [95,98,134]. ERM suggests finding the function that fits (e.g., minimizes the empirical error) a training set searching in a set of (possibly unknown [135–137]) set of functions $\mathcal{F}$ carefully tuned during the model selection phase [121]. The set of functions from which the ERM chooses depends on an accuracy term on the data, possibly plus one or more regularizers (e.g., p-norm of the weights) [134]. Regularizers can also be implicitly enforced via model functional form (e.g., convolutions or transformers) [98,134] or via optimization (e.g., stochastic gradient descent with early stopping and dropout) [134,138]. The tuning procedure trades off errors in the training data and the complexity of the solution. More formally, an ML algorithm that finds the best parameters θ, namely $f \in \mathcal{F}$, can be formulated as

$$f^* = \arg\min_{f \in \mathcal{F}} \mathtt{E}(f, \mathcal{D}_n) + \mathtt{C}(f), \tag{2}$$

where $\mathcal{F}$ is space of models induced, implicitly or explicitly, by the choice of the algorithms, $\mathtt{E}(f, \mathcal{D}_n)$ is the error, measured with reference to the desired metric [115], determined by the specific f over the data $\mathcal{D}_n$ and $\mathtt{C}(f)$ the complexity of the solution.

In order to select the best algorithm and hyperparameters configurations $\mathscr{A}^*_{H^*}$ in a set of possible ones $\mathcal{A}_H = \{\mathscr{A}^a_{H_1}, \dots, \mathscr{A}^a_{H_{h^a}}, \mathscr{A}^b_{H_2}, \dots, \mathscr{A}^b_{H_{h^b}} \cdots\}$ the following procedure has to be applied

$$\mathscr{A}^*_{H^*}: \quad \arg\min_{\mathscr{A}_H \in \mathcal{A}_H} \frac{1}{r} \sum_{i=1}^{r} \mathtt{E}(\mathscr{A}_H(\mathcal{L}^i_l), \mathcal{V}^i_v), \tag{3}$$

where $\mathscr{A}_H(\mathcal{L}^i_l)$ is a model f built with the algorithm $\mathscr{A}$ with its set of hyperparameters H and with the data $\mathcal{L}^i_l$, and where $\mathtt{E}(f, \mathcal{V}^i_v)$ is the desired metric for f computed using the data in $\mathcal{V}^i_v$. Since the data in $\mathcal{L}^i_l$ are independent of the data in $\mathcal{V}^i_v$, $\mathscr{A}^*_{H^*}$ should be an algorithm and the associated set of hyperparameters which allows achieving a small error on a data set that is independent of the training set.

Then, to evaluate the performance of the optimal model, which is $f^* = \mathscr{A}^*_{H^*}(\mathcal{D}_n)$ the following procedure has to be applied

$$\mathtt{E}(f^*) = \frac{1}{r} \sum_{i=1}^{r} \mathtt{E}(\mathscr{A}^*_{H^*}(\mathcal{L}^i_l \cup \mathcal{V}^i_v), \mathcal{T}^i_t). \tag{4}$$

Since the data in $\mathcal{L}^i_l \cup \mathcal{V}^i_v$ are independent from the ones in $\mathcal{T}^i_t$, $\mathtt{E}(f^*)$ is an unbiased estimator of the true performance, measured with the metric $\mathtt{E}$, of the final model [121].

Note that some ML algorithms can also leverage $\mathcal{U}_m$ to create the model, namely unlabeled samples. For this purpose, two main approaches exist

- if $\mathcal{D}_n$ is reasonably large, it is split in $\mathcal{L}_l$, $\mathcal{V}_v$, and $\mathcal{T}_t$ and then semi-supervised ML algorithms [106] are applied as follows

$$f^* = \mathscr{A}_H(\mathcal{L}_l \cup \mathcal{U}_m), \tag{5}$$

  namely, some algorithms can leverage both $\mathcal{U}_m$ and labeled samples to obtain a better model than simply leveraging the labeled samples. $\mathcal{U}_m$ is instead useless for performance tuning and assessing where labels are mandatory since errors need to be computed;
- if $\mathcal{D}_n$ is too small, splitting it into train, validation, and test is not possible. The $\mathcal{D}_n$ is split just in $\mathcal{V}_v$ and $\mathcal{T}_t$ (mandatory to be able to tune hyperparameters and assess the performance of the final model) and then unsupervised ML algorithms [139–141] are

[3] See also the top performing models in the popular Kaggle www.kaggle.com ML competition website.

[4] https://modelzoo.co

[5] https://huggingface.co

applied as follows

$$f^* = \mathscr{A}_H(\mathcal{U}_m), \tag{6}$$

namely, some algorithms can leverage $\mathcal{U}_m$ to create models that can be used in a supervised setting (e.g., anomaly and novelty detection).

Depending on the application, the needed accuracy, the type of input available, the experience of the data scientists, and the computational power, many different solutions may be found for better ZKPMs.

ZKPMs have different advantages and disadvantages. Regarding the advantages ZKPMs

- they require small if not zero intervention of the humans and time to be designed, constructed, and tested [47];
- they can fully exploit all the measures (inputs) in all the formats (unstructured and structured) that nowadays can be retrieved in a real application [47];
- if available data is large enough, they usually perform very well on average in real-world applications [47];
- they work very well in interpolation (i.e., close to the available measure or the design condition) [28];
- they usually require a very small amount of computational power when used to make predictions (with some exceptions for deep models or pre-trained/foundation models) [47].

Regarding the disadvantages ZKPMs

- they require a large amount of data to be constructed, huge in the case of deep models if no pre-trained/foundation model is available [39];
- they usually require a large (huge in the case of very deep models) amount of computational power to be constructed [47];
- they generally do not work well in extrapolation (i.e., far away from the available measure or the design condition) [28];
- since they are not guided by any knowledge of the physical phenomena they can be completely detached from the physical principles, sometimes generating completely absurd predictions [41]. In fact, they are designed to work well on average and not point-wise;
- they are, in general, not explainable and understandable. As a consequence, they are hard to be inspected, tested, and queried, even if some mitigation strategies exist [117,142–144].

Because of the reason explained above, ZKPMs are used in practice both like the FKPMs, namely for modeling purposes, but also to surrogate other, less computationally efficient, predictive models [40]. For example, complex computational fluid dynamics simulations that take days to make predictions can be surrogated with a ZKPM using previous simulations to generate predictions in some fraction of milliseconds. This is extremely important in practical applications where predictive models do not live just to make predictions, but are used for optimization or decision support.

3.3. Partial knowledge predictive models

PKPMs arise from acknowledgments of the advantages and disadvantages of both FKPMs (see Section 3.1) and ZKPMs (see Section 3.2) with the purpose of blending them toward the definition of predictive models able to take the best of the two worlds [40]. In particular, they try to fill the main limitation of FKPMs with respect to their computational demand in making accurate predictions and non-optimal performance, especially in interpolation, where instead ZKPMs excel, and the main limitation of ZKPMs with respect to their demand for a large amount of data, limited physical plausibility, and poor performance in extrapolation, where instead PKPMs excel [42].

For this reason, PKPMs require all four sources of information to be built (domain knowledge, measures, dataset, and experience) [72]. They are the newest way to build predictive models and are considered one of the most promising frontiers in industry and academia [72]. As with all $\mathtt{M}$, they are characterized by a model f chosen in a set of possible ones $\mathcal{F}$, and they can be designed as follows (see Fig. 11).

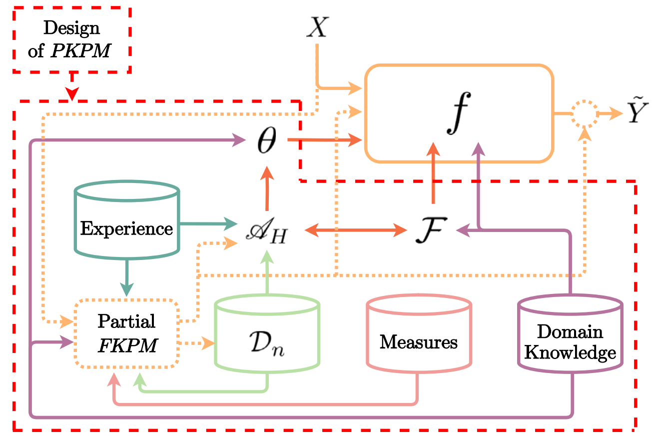

Fig. 11. PKPMs.

- usually in PKPMs, as for the ZKPMs, the predictive model f takes as inputs all the available ones X (with no problem in handling both quantity and variety). In general, as ZKPMs, there is a risk of stumbling upon a spurious correlation, but this is limited by the injection of domain knowledge with more or less complete FKPMs that inform the ZKPMs within three main ways
  - pre-processing (see Section 3.3.1): it refers to the preparatory steps taken to modify data, or guide the choice of the algorithm, before applying ZKPMs, aiming to enhance the models' ability to extract information effectively. This process is critically important in the context of ZKPMs due to the principle of "garbage in, garbage out", which emphasizes the quality of input data as a determinant of output quality. Pre-processing constitutes a method through which individuals, perhaps unknowingly, engage in informed ZKPMs practices such as feature engineering, data cleaning and cleansing, and data augmentation. These practices are vital in refining data to better suit ZKPMs, thereby improving model performance. For example, in serial PKPMs, an initial estimate is generated using FKPMs and then it is used as a hint for the subsequent ZKPMs, namely a prediction made by an FKPMs may be adjusted based on available data. Such methods fall into a category that leverages domain knowledge to assist ZKPMs in completing the "last mile", where human expertise may fall short. This approach underscores the symbiosis between domain-specific insights and ZKPMs, aiming to bridge the gap where human knowledge alone is insufficient, thereby enhancing the model's predictive accuracy and reliability;
  - in-processing (see Section 3.3.2): it involves integrating domain knowledge into the core learning mechanism ZKPMs, thereby infusing it directly into the ZKPMs learning process. This approach necessitates encoding domain knowledge into a more mathematical form, such as laws, trends, or constraints and requires robust collaboration between domain experts and data scientists. Unlike pre-processing, in-processing demands modifications to the mathematical learning mechanism itself, aiming to steer the model toward a desired behavior. Such modifications may involve altering the functional structure of the model, introducing constraints, or incorporating new regularization techniques. Specific techniques like knowledge distillation [145,146], where a simpler model tries to approximate a more complex one, can be used in this context to distillate a simple ZKPM from a complex FKPM. These adjustments are made to preserve the advantageous properties of ZKPMs, such as

convexity and differentiability while ensuring that domain knowledge contributes effectively to the model's performance. Moreover, this strategy seeks to enhance the model's point-wise performance, not merely its average ones, by making it capable of leveraging domain-specific insights. This approach underscores the importance of a synergistic relationship between domain expertise and data science, enabling the development of more accurate, reliable, and context-aware ZKPMs;
– post-processing (see Section 3.3.3): it involves adjusting the outputs of an ML model to align with domain-specific knowledge after the model has been applied. One method involves using ZKPM predictions to estimate quantities, which are then utilized within FKPMs to enhance explainability and reduce possible errors (e.g., parallel gray box models). Another approach entails enforcing the ZKPMs outputs to adhere to specific logical rules. For instance, if the model determines that an obstacle is a pedestrian, it cannot simultaneously decide to continue at the same speed and trajectory, implying that the path is clear. Generally, like pre-processing, this approach aims to leverage the ZKPMs "as-is" capitalizing on pre-existing functionalities without directly altering the ZKPMs mechanism itself. However, similar to in-processing, it necessitates significant manipulation of the predictor's behavior to mitigate potential errors inherent in ZKPMs outputs, requiring profound domain knowledge. This strategy emphasizes the importance of integrating domain expertise with ZKPMs outputs to ensure that the final results are both accurate and contextually relevant; By applying domain-specific constraints and knowledge post-prediction, researchers and practitioners can refine the utility and applicability of ZKPMs, particularly in complex or sensitive areas where logical consistency and domain fidelity are of paramount importance. This approach underscores the need for a collaborative effort between domain experts and data scientists to enhance the interpretability and reliability of machine learning systems;

also note that there is not always such a clear cut between the approaches to define a PKPM and that pre-in- and post-processing can be combined;
- since PKPMs combine FKPMs and ZKPMs, they are characterized by the parameters of the FKPMs and parameters and hyperparameters of the ZKPMs that are tuned as explained in Sections 3.1 and 3.2 respectively;

PKPMs have different advantages and disadvantages that can be summarized as follows
- they can handle all kinds of inputs and outputs and large amounts of data as for ZKPMs [41];
- they can encapsulate different levels of domain knowledge as they can leverage also partial knowledge that would not be enough to generate an FKPMs [72];
- they require less data than ZKPMs but more data than FKPMs [40];
- they require less human intervention than FKPMs but more than ZKPMs [68];
- they require a large amount of computational power (and then time) to be constructed but surely less than FKPMs (that require human time and not simply computational power) and usually a bit more than ZKPMs [41];
- they are extremely computationally cheap to use to make predictions as ZKPMs [47];
- they perform very well both in interpolation (better than FKPMs and sometimes even better than ZKPMs) and extrapolation (often reaching if not surpassing the performance of FKPMs) [41];
- they are usually much more physically plausible than ZKPMs but not always reaching the same level of physical plausibility as FKPMs [41];

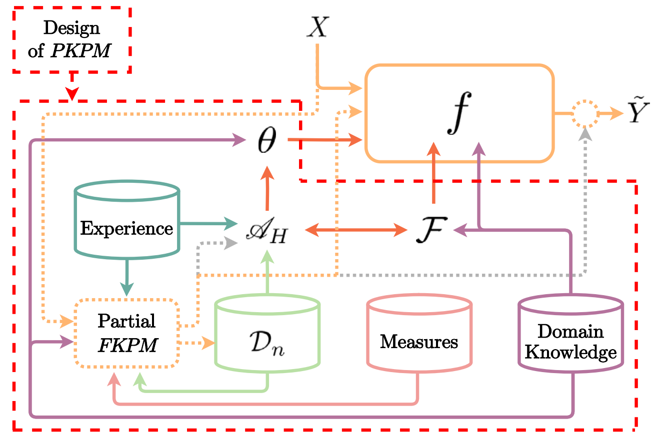

Fig. 12. PIPMs: pre-processing.

- they work well, on average, as ZKPMs, and they often reach if not surpass the point-wise performance of FKPMs [58];
- they improve the explainability of ZKPMs, but they often remain quite unexplainable as ZKPMs, especially when they are asked to leverage a large mass of heterogeneous data [48];

3.3.1. Pre-processing

In PKPMs, pre-processing plays a crucial role in enhancing the predictive model performance by leveraging domain knowledge (i.e., a potentially partial FKPM) to effectively inform, a-priori, ZKPMs (see Fig. 12 where we took the general schema of PKPMs in Fig. 11 and we put in gray the parts that are not relevant for pre-processing PKPMs). Pre-processing acts, a-priori, in different ways
- modifying the input X [147];
- modifying the dataset $\mathcal{D}_n$ [54];
- guiding the ZKPMs selection $\mathscr{A}_H$.

Modifying the input means cleaning and cleansing the inputs to remove inconsistencies and errors, ensuring that the information fed into the ZKPMs is of high quality [52]. Also, feature engineering and enrichment (i.e., transforming existing data attributes and creating new features to capture the underlying physical phenomena) are relevant to the task [147] . Feature reduction or selection techniques to minimize redundancy and focus on the most informative attributes, thereby optimizing computational efficiency and model interpretability, are also important steps in modifying the inputs [51]. More formally, this step can be formulated as

$$X \to \phi(X), \tag{7}$$

where $\phi : \mathcal{X} \to \Phi$ is a function that maps the input from the original space to another one induced by the domain knowledge (i.e., a potentially partial FKPMs).

Modifying the dataset involves selecting and enriching the available data, not only choosing the most appropriate data but also designing experiments to collect this data if necessary, ensuring it accurately represents the phenomena under study [148]. This might involve recognizing which subsets of data are of higher quality due to external factors [149]. Data enrichment strategies are similar to the analogy of rotating images in image recognition or using FKPMs to generate more data [54]. More formally, this step can be formulated as

$$\mathcal{D}_n \to \mathcal{D}'_{n'}, \tag{8}$$

namely, we modify and transform the dataset (removing, stratifying in quality, and adding samples).

Guiding the ZKPMs selection means guiding the selection of the ML algorithm (or hyperparameters) behind the ZKPMs [72]. This is a pivotal part of the process and should be informed by the specific

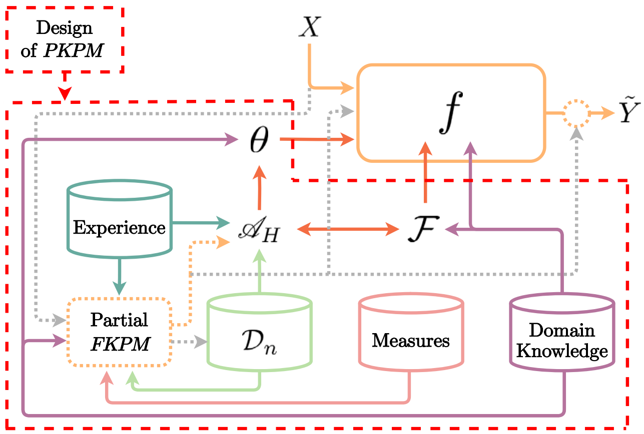

Fig. 13. PIPMs: in-processing.

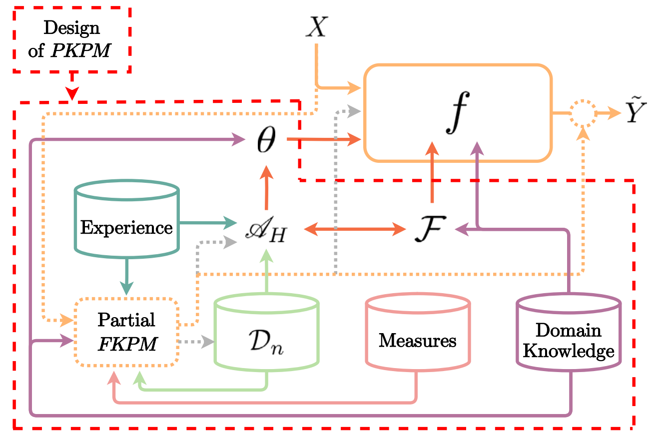

Fig. 14. PIPMs: post-processing.

characteristics of the pre-processed and enriched dataset and the physical laws and domain knowledge underpinning the problem space [72]. For example, if we are in a safety-critical situation, it is better to use (even if less accurate) a fully interpretable model [143]. If we deal with images, convolutions are the best choice, while transformers are the way to go if we have to deal with natural language [39]. If we have a lot of structured data, deep models are probably the best choice, while for medium/small cardinality unstructured datasets, shallow models are the optimal choice [95,98]. This mix of domain knowledge and experience can make a difference in delivering PKPMs. More formally, this step can be formulated as

$$\mathscr{A}_H \to \mathscr{A}'_{H'}, \tag{9}$$

namely, we modify and transform the choices to be made in the performance assessment phase.

This comprehensive approach to pre-processing prepares the data for the learning process and aligns it closely with the physical insights and constraints of the domain, setting the stage for more accurate, reliable, and interpretable PKPMs.

3.3.2. In-processing

In PKPMs, in-processing plays a crucial role in enhancing the predictive model performance by leveraging domain knowledge of a potentially partial, but well mathematically encoded, FKPMs to inform the learning process of a ZKPMs effectively (see Fig. 13 where we took the general schema of PKPMs in Fig. 11 and we put in gray the parts that are not relevant for in-processing PKPMs).

In-processing in PKPMs involves the integration of domain-specific physical laws and principles directly into the model training process to ensure that the models adhere closely to known scientific knowledge [41]. This means acting on the model's functional form and in the way its parameters are actually found. Selecting the functional form means selecting how inputs are translated in predictions (e.g., decision trees, boosting, bagging, convolutions, and attentions) [47]. When certain physical laws are known, these can be used to guide the model in several ways [72]. One approach is to ensure the model's predictions do not deviate significantly from these laws by incorporating a regularization term that penalizes deviations from the expected physical behavior [41]. This regularization can help maintain the model's fidelity to the physical law, such as the relationship between specific inputs and outputs, ensuring that if an input value increases, the output adjusts in a physically consistent manner [72]. Moreover, when dealing with complex systems where simulators exist but are too slow for practical use, surrogate models can be developed [55–58]. These models aim to mimic the simulator's outputs while being computationally efficient, thus requiring the model to accurately capture the underlying physical relationships [55–58]. Incorporating physical laws into machine learning does not always require exhaustive or precise details; even hints or partial knowledge about the physical system can be beneficial [72]. This approach allows the inclusion of soft constraints or priors that guide the learning process, ensuring that the model's predictions remain plausible within the physical context [72]. During performance tuning, preference can be given to models that not only achieve comparable accuracy but also align more closely with domain knowledge, are simpler, more explainable, and adhere to the known physical laws [72]. This preference toward models that respect domain knowledge ensures that the final PKPM solutions are not just data-driven but also grounded in the physical reality of the domain, enhancing their reliability, interpretability, and applicability to real-world problems [41]. More formally, this step can be formulated as a modification of the learning process of ZKPMs as formulated in Eq. (2)

$$\mathcal{F} \to \mathcal{F}', \tag{10}$$

$$f^* = \arg\min_{f\in\mathcal{F}'} \mathtt{E}(f,\mathcal{D}_n) + \mathtt{C}(f) + \mathtt{K}(f), \tag{11}$$

namely, we modify and transform the function space $\mathcal{F}$ and we add a regularizer $\mathtt{K}(f)$ that embeds domain knowledge [58].

In summary, the in-processing strategy within PKPMs represents a paradigm shift toward developing ML models that are informed by and compliant with the foundational principles of physical sciences [72]. This approach significantly contributes to creating models that are not only predictive but also interpretative and aligned with the real-world phenomena they aim to simulate, thereby bridging the gap between data-driven insights and physical domain knowledge [72].

3.3.3. Post-processing

In PKPMs, post-processing leveraged domain knowledge (i.e., potentially, partial bust still well mathematically encoded, FKPMs) to align the output of ZKPMs (see Fig. 14 where we took the general schema of PKPMs in Fig. 11 and we put in gray the parts that are not relevant for post-processing PKPMs).

Post-processing in PKPMs focuses on refining predictions of ZKPMs to ensure they align with domain-specific knowledge, constraints, and practical considerations [104]. This stage is crucial for enforcing certain characteristics and relationships that must be present in the predictions [104]. For instance, in a monitoring system, if a model predicts the health status of machinery, it must recognize hard constraints such as a fault prediction of a piece of the system means that there is an anomaly also for the entire system (i.e., constraint in the hierarchy of the predictions). Furthermore, post-processing involves adjusting predictions to ensure they do not deviate excessively from established practices, like the settings of machinery, thereby ensuring that the model's recommendations are practical and implementable within current protocols [72]. This step not only enhances the model's reliability but also its acceptance among practitioners [104]. Ensuring that the model's decisions are in harmony with domain knowledge can be straightforward if the model is inherently explainable [117]. For models that are not easily interpretable, techniques such as feature importance analysis can provide global explainability, offering insights into what the model considers important across all decisions [150,151]. For local explainability–understanding individual predictions–tools can be employed to break down the decision-making process for specific cases [150]. Finally, one can decide to build FKPMs, then interpretable and controllable, that require some inputs that are not easy to measure or estimate: in this case, ZKPMs may be of support and more easily controllable since their estimate can be verified and controlled by the subsequent FKPMs toward final PKPMs [40]. The simplest way is to use a ZKPM to adjust the prediction of a FKPM, using the so-called parallel hybrid models architecture [60]. More formally, this step can be formulated as a modification of the output of the ZKPMs

$$f(X) \to \psi(f(X)) \tag{12}$$

namely, we modify the output by means of a function $\psi : \mathcal{Y} \to \mathcal{Y}$ that encapsulates our domain knowledge.

The good property of the post-processing approach lies in its ability to adapt and apply ZKPMs as is, without needing to modify the underlying, maybe already running, predictive models [40]. By applying these post-processing steps, practitioners can ensure that the predictions are not only accurate but also adhere to physical laws, ethical considerations, and practical necessities, making them more relevant and trustworthy in real-world applications [48]. This approach leverages the power of ZKPMs while respecting the nuances and constraints of the physical and practical world [40].

4. Illustrative examples

In this section, we will present the application of the concepts illustrated in Section 3 to a simple example to better understand what has been presented before, how to apply it in practice, and what results one could expect.

Specifically, Section 4.1 will present our illustrative example: a mass–spring-damper system with no external force applied. Then, Section 4.2 will present how data has been generated. Subsequently, Section 4.3 will present the application of the FKPMs, the ZKPMs, and the PKPMs to our illustrative example. Then, the scenario under consideration (Section 4.4) and the evaluation methods (Section 4.5) will be presented. Finally, Section 4.6 will report the results and discuss them. All the code developed to derive the results of this section has been made publicly available.[6]

4.1. The mass-spring-damper system

Let us start with the description of the illustrative example. Let us consider a mass–spring-damper system [152], with no external force applied, represented in Fig. 15. The mass–spring-damper system is a model used in physics and engineering to describe a simple harmonic oscillator with damping. It consists of three primary components: a mass (m) representing the oscillating object, a spring (k) attached to the mass that provides a restoring force that tries to bring the mass back to its equilibrium position, and a damper (μ) that models the resistance or damping force that opposes the motion of the mass. In the absence of external forces, the mass is initially positioned at a specific point, following which the system is allowed to evolve autonomously over time.

[6] https://github.com/lucaoneto/PhysicsInformedMassSpringDamper

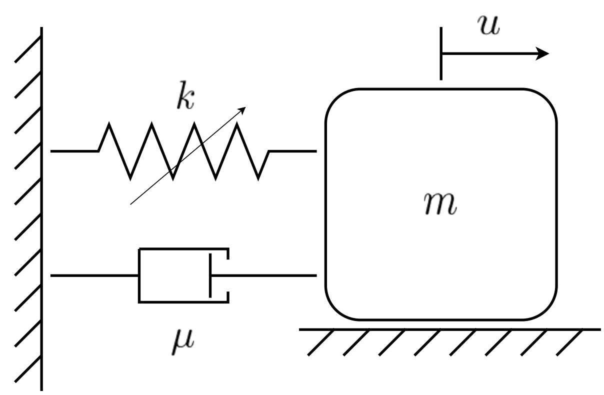

Fig. 15. Mass-spring-damper system.

Assume we have the capability to measure the displacement $u(t)$ of the mass at various moments in time $t \in [0, t_m]$. The scope is to predict the mass's position at any given time (both in the past, $t \in [0, t_m]$, and in the future $t \in (t_m, t_f])$.

We will call our measures, according to what described in Section 3, dataset, i.e., $\mathcal{D}_n = \{(x_1, y_1), \dots, (x_n, y_n)\} = \{(t_1, u(t_1)), \dots, (t_n, u(t_n))\}$, namely our input space is the time and our output space is the mass's position.

The FKPMs that we can formulate in this scenario are numerous and depend on the level of accuracy that we want to achieve. The simplest FKPM is commonly described by a second-order differential equation [152]

$$m\frac{d^2u(t)}{dt^2} + \mu\frac{du(t)}{dt} + k_0 u(t) = 0. \tag{13}$$

where

- $m\frac{d^2u(t)}{dt^2}$ represents the inertial force of the mass m (measured in $[kg]$), and $\frac{d^2u(t)}{dt^2}$ denoting the acceleration;
- $\mu\frac{du(t)}{dt}$ corresponds to the damping force and $\frac{du(t)}{dt}$ to the velocity. μ (measured in $[{}^{Ns}\!/\!_{m}]$) and $\frac{du(t)}{dt}$ together model the damping resistance or frictional force;
- $k_0 u(t)$ represents the restoring force exerted by the spring, with k_0 (measured in $[{}^{N}\!/\!_{m}]$) as the spring constant and $u(t)$ as the displacement from equilibrium.

For scenarios where the linear approximation of the proportional relationship between force and displacement of the spring is insufficient, a more complex representation can be considered [152]

$$m\frac{d^2u(t)}{dt^2} + \mu\frac{du(t)}{dt} + k_1 u(t) + k_2 u^3(t) = 0. \tag{14}$$

where, as in Eq. (13), $k_1 u(t)$ represents the linear restoring force exerted by the spring, $k_2 u^3(t)$ approximates the nonlinear restoring forces, and k_1 (measured in $[{}^{N}\!/\!_{m}]$) and k_2 (measured in $[{}^{N}\!/\!_{m^3}]$) are the constants that quantify the strength of the linear and nonlinear effects respectively.

Note that other, more accurate, FKPMs can be formulated (e.g., inserting a non-linearity in damping force), but this is out of the scope of this section. What is important, for the scopes of this section, is that the second FKPM of Eq. (14) is more accurate than the first FKPM of Eq. (13).

4.2. Data generations

Based on what was described in the previous section, we now present how the $\mathcal{D}_n$ is generated. Specifically, we consider two aspects

- the data-generating model;
- the quality of the measures.

For what concerns the data-generating model, we will consider two FKPMs:

Table 2
Data generation parameters.

Parameter	Eq. (13)		Eq. (13)
m		1	
u_0		1	
t_f		1	
t_m		$^{t_f}/_3$	
σ		.1	
μ		3	
k_0	250		–
k_1	–		230
k_2	–		3

- the most accurate one of Eq. (14), that, for the purpose of this section, will be considered as the oracle (i.e., the ground truth). In other words, this represents the real system;
- the less accurate one of Eq. (13), that, for the purpose of this section, will be considered as a good approximation of Eq. (14). In other words, this represents a good simulator of the real system or a surrogate of the real system using an FKPM.

For what concerns the quality of the measures, we will consider two scenarios:

- a theoretical one where we assume to be able to measure exactly the time and position of the mass;
- a realistic one where the measures are corrupted by noise.

Then, in order to generate our data, we simply define an initial displacement $u(0) = u_0$, and then we solved Eqs. (13) and (14) using the Eulero's method [153] obtaining $u(t)$, and then we generated the following 4 datasets

- $\mathcal{D}_n^{(13),0} = \{(t, u(t))|t \in \{0, \frac{t_m}{n-1}, \dots, t_m\}\}$ with $u(t)$ the solution of Eq. (13);
- $\mathcal{D}_n^{(13),\sigma} = \{(t, u(t) + \mathcal{N}(0, \sigma^2))|t \in \{0, \frac{t_m}{n-1}, \dots, t_m\}\}$ with $u(t)$ the solution of Eq. (13);
- $\mathcal{D}_n^{(14),0} = \{(t, u(t))|t \in \{0, \frac{t_m}{n-1}, \dots, t_m\}\}$ with $u(t)$ the solution of Eq. (14);
- $\mathcal{D}_n^{(14),\sigma} = \{(t, u(t) + \mathcal{N}(0, \sigma^2))|t \in \{0, \frac{t_m}{n-1}, \dots, t_m\}\}$ with $u(t)$ the solution of Eq. (14);

where $\mathcal{N}(0, \sigma^2)$ is a realization of a Gaussian distribution of mean 0 and variance σ^2, and n is the number of samples.

In our data generation process the free parameters have been set as reported in Table 2.

In order to give a better idea of the FKPMs of Eqs. (13) and (14) and the generated data ($\mathcal{D}_n^{(13),0}$, $\mathcal{D}_n^{(13),\sigma}$, $\mathcal{D}_n^{(14),0}$, and $\mathcal{D}_n^{(14),\sigma}$) for a single realization of the noise, we report them in Fig. 16.

4.3. Full-, zero-, and partial-knowledge predictive models

In this section we will describe how to build FKPMs, ZKPMs, and PKPMs for our specific illustrative example. Note that our proposal is not the only possible one. Nevertheless, to the authors' best knowledge, it is the state-of-the-art one for each case.

In a real scenario, the available information is:

- the domain knowledge about the problem; In this case the physical knowledge about the system of Fig. 15;
- measures (the mass, as we will see later);
- the available data; In this case $\mathcal{D}_n^{(13),0}$, $\mathcal{D}_n^{(13),\sigma}$, $\mathcal{D}_n^{(14),0}$, or $\mathcal{D}_n^{(14),\sigma}$;
- the experience of the authors.

To develop the FKPM, as a first step (see Section 3.1), we need to start from the physical knowledge and generate the functional form of the FKPM. The result of this process can be, of course, tautologically, the two FKPM of Eqs. (13) and (14) that we use to generate the data since this is an illustrative example. The difference, in this part, is that, since we have just the physical knowledge and the data available, some parameters of those models can be assumed to be known (e.g., m since it can be easily measured statically or it can be known in advance) while some others need the be tuned on the available data (e.g., u_0, μ, k_0, k_1, and k_2 since the only quantity that we assume that we can measure about the dynamic system is the time and position). A simple way to tune the unknown parameters is to grid search them, finding the parameters that best fit the available data. In other words, one needs to test multiple combinations of the unknown parameter, solve Eqs. (13) or (14) based on the opted FKPM using the Eulero's method and then select the combination with minimal error on $\mathcal{D}_n^{(13),0}$, $\mathcal{D}_n^{(13),\sigma}$, $\mathcal{D}_n^{(14),0}$, or $\mathcal{D}_n^{(14),\sigma}$. In our case, we searched the parameters in the $[1, 100]$ range with a grid of 45 points equally spaced in logarithmic scale (15 point each decade).

To develop the ZKPM, as a first step (see Section 3.2), we need to choose the right ML algorithms and then tune its hyperparameters just based on the available data with a grid search hold-out method. Of course, one should test multiple algorithms but in practice, based on the experience of the data scientist or machine learner, it can be immediately clear from the beginning a good algorithm to test (e.g., a universal approximator that has shown in the past to work well in similar problems). In our case, we proceeded as follows. First of all, note that the input x (in our case the time) is in $[0, 1]$ ($t = {}^t/_{t_f}$). Then, we defined a functional form of the model as a polynomial of degree p (a universal approximator)

$$f(x) = \sum_{i=0}^{p} w_i x^i, \tag{15}$$

where p is a hyperparameter and $w_0, \dots, w_p$ are parameters. Then, as loss functions, we opt for the squared error

$$\ell(f(x), y) = (y - f(x))^2. \tag{16}$$

Then, as regularization, we opt for a classical measure of complexity [154]

$$\mathrm{C}(f) = \int_0^1 \left(\frac{d^2 f(x)}{dx^2} \right)^2 dx = \boldsymbol{w}' C \boldsymbol{w}, \tag{17}$$

where

$$C_{i,j} = \begin{cases} 0 & \text{If } j < 2 \vee j < 2 \\ \frac{i(i-1)j(j-1)}{i+j-3} & \text{Otherwise} \\ , & \end{cases} \quad i, j \in \{0, \dots, p\}. \tag{18}$$

Then, our algorithm will be (see the relation with Eq. (2) in Section 3.2)

$$\min_{\boldsymbol{w}} \quad \|X\boldsymbol{w} - \boldsymbol{y}\|^2 + \lambda \boldsymbol{w}' C \boldsymbol{w}, \tag{19}$$

where

$$\boldsymbol{w} = \begin{bmatrix} w_0 \\ \vdots \\ w_p \end{bmatrix}, \quad X = \begin{bmatrix} x_1^0 & \cdots & x_1^p \\ \vdots & \ddots & \vdots \\ x_n^0 & \cdots & x_n^p \end{bmatrix}, \quad \boldsymbol{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}, \tag{20}$$

and λ is a hyperparameter. Note that Problem (19) has a closed form solution

$$\boldsymbol{w} = (X'X + \lambda C)^{+} X' \boldsymbol{y}, \tag{21}$$

where $(X'X + \lambda M)^+$ is the Moore–Penrose inverse of $(X'X + \lambda M)$. In order to tune our hyperparameters (in our case p and λ), we can use multiple grid search hold-out methods (e.g., cross-validation and bootstrap). Given the low cardinality of n (see next section) we will use the leave-one-out method. In our case, we searched the λ hyperparameter in the range $[10^{-6}, 10^{+6}]$ with a grid of 65 points equally spaced in

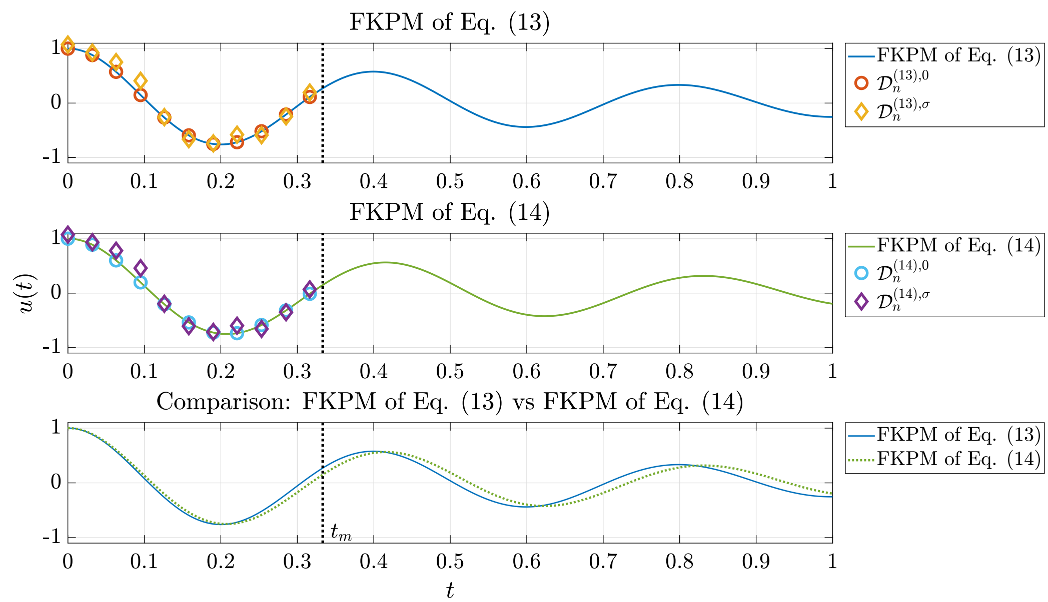

Fig. 16. FKPMs of Eqs. (13) and (14) and the generated data ($\mathcal{D}_n^{(13),0}$, $\mathcal{D}_n^{(13),\sigma}$, $\mathcal{D}_n^{(14),0}$, and $\mathcal{D}_n^{(14),\sigma}$) for a single realization of the noise.

logarithmic scale (5 point each decade) and the p hyperparameter in $[0, 1, \dots, 30]$.

Finally, to develop the PKPM (see Section 3.3), we need to combine the previous FKPMs and ZKPMs. In fact, in this case, we will fully leverage the two sources of information: physical knowledge and the data. Note that, as described in Section 4.2, we consider the FKPM of Eq. (14) as a sort of oracle that in practice is unknown, while the FKPM of Eq. (13) is known also in practice. This means that our PKPM needs both to fit the data as the ZKPM and also exploit the physical information embedded in the FKPM of Eq. (13). The state-of-the-art approach to achieve this goal is to add a penalty term in the ZKPM of Eq. (19) that forces the $f(x)$ (namely the approximation of $u(y)$) to follow the principle of Eq. (13). More formally

$$\mathrm{P}(f) = \int_0^1 \left(m\frac{d^2 f(x)}{dx^2} + \mu\frac{d f(x)}{dx} + k_0 f(x) \right)^2 dx = \boldsymbol{w}' P \boldsymbol{w}, \tag{22}$$

where

$$P = k_0^2 P_1 + \mu^2 P_2 + m^2 C + k_0\mu(P_3 + P_3') + \mu m(P_4 + P_4') + k_0\ m(P_5 + P_5'), \tag{23}$$

with

$$P_{1,i,j} = \frac{1}{i+j+1}, \quad P_{2,i,j} = \begin{cases} 0 & \text{If } i < 1 \vee j < 1 \\ \frac{ij}{i+j-1} & \text{Otherwise} \end{cases},$$

$$P_{3,i,j} = \begin{cases} 0 & \text{If } i < 1 \\ \frac{i}{i+j} & \text{Otherwise} \end{cases}, \quad P_{4,i,j} = \begin{cases} 0 & \text{If } i < 2 \vee j < 1 \\ \frac{i(i-1)j}{i+j-2} & \text{Otherwise} \end{cases},$$

$$P_{5,i,j} = \begin{cases} 0 & \text{If } i < 2 \\ \frac{i(i-1)}{i+j-1} & \text{Otherwise} \end{cases}, \tag{24}$$

where $i, j \in \{0, \dots, p\}$. Note that the quantity of Eq. (22) forces the model $f(x)$ to follow the FKPM of Eq. (13) inexpensively with respect to other naive approaches, such as sampling different values of x and impose the point-wise equality. Thanks to Eq. (22) we can modify the ZKPM of Eq. (19) as follows (see the relation with Eq. (11) in Section 3.3 and more specifically, Section 3.3.2)

$$\min_{\boldsymbol{w}} \quad \|X\boldsymbol{w} - \boldsymbol{y}\|^2 + \lambda_1 \boldsymbol{w}' C \boldsymbol{w} + \lambda_2 \boldsymbol{w}' P \boldsymbol{w}, \tag{25}$$

where p, λ_1, and λ_2 are the hyperparameters of the algorithm. Note that Problem (25) has the following closed form solution

$$\boldsymbol{w} = (X'X + \lambda_1 C + \lambda_2 P)^{+} X' \boldsymbol{y}. \tag{26}$$

In order to tune our hyperparameters (p, λ_1, and λ_2) we will use, as for the ZKPM, the leave-one-out method searching λ_1 and λ_2 in the range $[10^{-6}, 10^{+6}]$ with a grid of 65 points equally spaced in logarithmic scale (5 point each decade) and the p hyperparameter in $[0, 1, \dots, 30]$. As a final remark, note that the value of k_0 and μ need to be set based on the scenario under consideration (see Section 4.4), as described in Section 3.3, while m, as described for the FKPM, can be measured.

4.4. Scenarios: Surrogation and modeling

In this section, we will describe the two realistic scenarios we considered to test the performance of the FKPM, ZKPM, and PKPM performance described in the previous section. In particular, we considered two different scenarios: the surrogation and the modeling.

In the surrogation scenario, we consider the case having a simulator of the reality (in our case the FKPM of Eq. (13)) which can be a software simulator (and then we can measure everything with no noise producing $\mathcal{D}_n^{(13),0}$) or a model scale test (and then we can make measure corrupted by noise producing $\mathcal{D}_n^{(13),\sigma}$) that we want to surrogate since it takes too much time to make predictions. Then, starting from $\mathcal{D}_n^{(13),0}$ or $\mathcal{D}_n^{(13),\sigma}$ we want to fit the very same FKPM (i.e., the one described in Section 4.3, to test how good would be to tune the very same model on the data produced by its specific instance), the ZKPM of Section 4.3, and the PKPM of Section 4.3. Note that, in this scenario of PKPM, k_0 and μ are known exactly both if we are using a software simulator and both if we are performing a model scale test (since the physical simulator can be manipulated arbitrarily).

In the modeling scenario, instead, we consider the case of measuring data from the real system (in our case the FKPM of Eq. (14)) and to be able to measure time and displacement with very high precision producing $\mathcal{D}_n^{(14),0}$ or to have lower quality measures producing $\mathcal{D}_n^{(14),\sigma}$. Then, starting from $\mathcal{D}_n^{(14),0}$ or $\mathcal{D}_n^{(14),\sigma}$ we want to fit a simpler FKPM (i.e., the one described in Section 4.3, simulating a partial physical knowledge of the reality), the ZKPM of Section 4.3, and the PKPM of Section 4.3. note that, in this scenario of PKPM, both k_0 and μ are unknown parameters. To determine their values, we will apply an FKPM fitting process to the data to extract these quantities.

Table 3
Results for the Surrogation scenario using $\mathcal{D}_n^{(13),0}$: training time, prediction time, interpolation error, and extrapolation error of FKPM (with $n = 11$), ZKPM, and PKPM (with $n = 22$).

		FKPM	ZKPM	PKPM
		($n = 11$)	($n = 22$)	
MAE	Interpolation	0.0176	0.00179	0.00143
	Extrapolation	0.0374	2.888	0.0253
Time	Training	~ 160 s	~ 5 s	~ 170 s
	Predicting	$\sim 10^{-3}$ ms	$\sim 10^{-4}$ ms	$\sim 10^{-4}$ ms

4.5. Evaluation: Interpolation and extrapolation

This section will describe how we will evaluate the performance of the different models (FKPM, ZKPM, and PKPM) in the scenarios (surrogation and modeling).

For this purpose, we will use tree metrics: the training time, the prediction time, and the accuracy.

For the training time, we intend the time needed to construct the FKPM, ZKPM, and PKPM using the available data (we disregard the time needed to acquire that physical knowledge for the FKPM).

For the prediction time, given a specific time t, we intend the time needed to predict the displacement $u(t)$.

For accuracy, the discussion is a bit more complicated. As described in Section 4.2 and also looking at Fig. 16, data are sampled for $t \in [0, t_m = {}^{t_f}/_3]$ while phenomena is observer for $t \in [0, t_f = 1]$. If we examine the error of our predictive model within the time interval $t \in [0, t_m]$, we are essentially analyzing the interpolation error. This refers to the deviation or discrepancy of the model's predictions around the data points we already have. If we look at the error of our predictive model in $t \in [t_m, t_f]$, we are looking at the extrapolation error, the error far away from the available data, namely in conditions non-observed during the data collection. As an error measure, we will exploit the Mean Absolute Error (MAE) between the data generating model (i.e., the FKPM of Eq. (13) for the surrogation scenario and the FKPM of Eq. (14) for the modeling scenario).

4.6. Results and discussion

In this section, we will present the results of our illustrative example of Section 4.1 of applying the methodology described in Section 4.3 on the scenarios described in Section 4.4 evaluating the results according to Section 4.5 leveraging the data described in Section 4.2.

In order to compare FKPM, ZKPM, and PKPM fairly, the cardinality of the dataset n has been set differently for the FKPM with respect to the ZKPM and the PKPM. In fact, as explained in Section 3, FKPM is usually exploited when the number of samples is limited while ZKPM and PKPM are usually exploited when a larger number of samples is available. In our case we set $n = 11$ for the FKPM and $n = 22$ for the ZKPM and the PKPM.

According to what was described in the previous sections, we will report four different conditions

- surrogation scenario (we have a simulator of the reality, i.e., the FKPM of Eq. (13))
  - with no noise $\sigma = 0$ (we have a software simulator, i.e., we can measure everything with no noise, producing $\mathcal{D}_n^{(13),0}$). The results of this case are reported in Table 3 and Fig. 17;
  - with noise (see Table 2) $\sigma = 0.1$ (we have a model scale test, i.e., we can make measures corrupted by noise, producing $\mathcal{D}_n^{(13),\sigma}$). The results of this case are reported in Table 4 and Fig. 18;
- modeling scenario (we measure data from the real system, i.e., the FKPM of Eq. (14))
  - with no noise $\sigma = 0$ (we measure time and displacement with very high precision producing $\mathcal{D}_n^{(14),0}$). The results of this case are reported in Table 5 and Fig. 19;
  - noise $\sigma = 0.1$ (we have lower quality measures producing $\mathcal{D}_n^{(14),\sigma}$). The results of this case are reported in Table 6 and Fig. 20.

Specifically, Tables 3–6 report, the MAE (in interpolation and extrapolation) and the Time (training and test) of the tree predictive models (FKPM, ZKPM, and PKPM), averaged over 30 repetition of the experiments in case of the presence of the noise (in order to remove the dependency of the results from the single noise realization). Figs. 17–20, instead, report the Ground Truth (the FKPM of Eq. (13) for the surrogation scenario and the FKPM of Eq. (14) for the modeling scenario), the predictive models (FKPM, ZKPM, and PKPM), and the datasets leveraged to build the predictive models. In this case, we reported just a single representative realization of the noise in the cases of noise presence, for space constraints and for clarity.

From Tables 3–6 and Figs. 17–20 it is possible to observe that:

- the FKPMs usually behave quite well both in interpolation and in extrapolation, producing predictions that are consistent with the physical behavior of the system;
- the ZKPMs outperform the FKPMs in interpolation, while in extrapolation, the ZKPMs predictions are completely meaningless with respect to the physical behavior of the system;
- the PKPMs outperform the FKPMs and the ZKPMs in both interpolation and extrapolation, producing both accurate and physically consistent predictions;
- result does not change in the presence or absence of noise. The presence of noise simply reduces the performance of both FKPMs, ZKPMs, and PKPMs;
- the training time is high for all the models. The time depends mostly on the granularity of the grid for the grid search (see Section 4.3). One could optimize this search, but this is out of our scope. Note that the PKPMs is the most computationally demanding predictive model to be built since it requires building an FKPM and a ZKPM;
- the prediction time is very low for all models since our illustrative example is very simple. Nevertheless, one can still observe an improvement of one order of magnitude in prediction time using the ZKPMs or PKPMs instead of FKPMs;

note that these results matched the theoretical expectation discussed in Section 3.

Finally, note that increasing the granularity of the grid search could improve the quality of the results. For example, a very fine grid in the case of the absence of noise for the surrogate scenario may strongly increase the quality of the FKPM which will tend to zero. Nevertheless, this is not feasible in a practical situation, resulting in an explosion of the training time.

5. Full-, zero-, and partial-knowledge predictive models for industrial applications

This section provides the actual review of FKPMs, ZKPMs, and PKPMs for industrial applications focusing, as described in the introduction, on six main domains: extraction (Section 5.1), chemical (Section 5.2), manufacturing (Section 5.3), transportation (Section 5.4), energy (Section 5.5), and construction (Section 5.6).

For this purpose, as in Section 2, we searched for papers in a specific time window using a series of keywords in the academic database

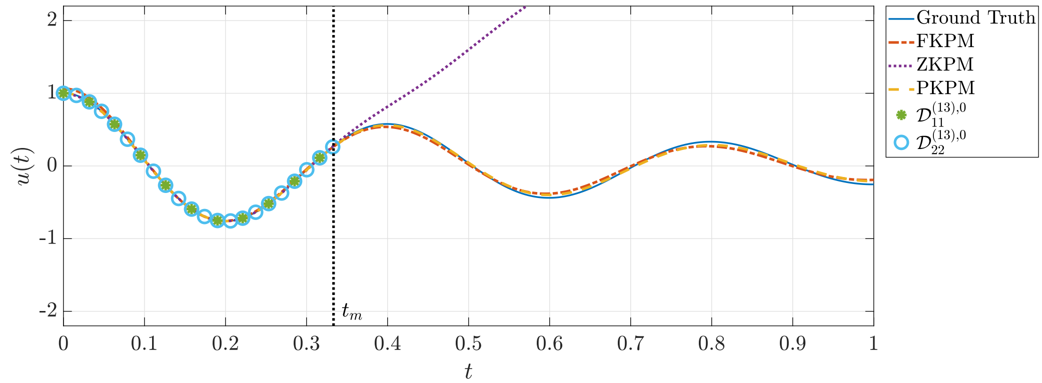

Fig. 17. Results for the Surrogation scenario using $\mathcal{D}_n^{(13),0}$: ground truth, $\mathcal{D}_{11}^{(13),0}$, $\mathcal{D}_{22}^{(13),0}$, and the prediction of the FKPM, ZKPM, and PKPM.

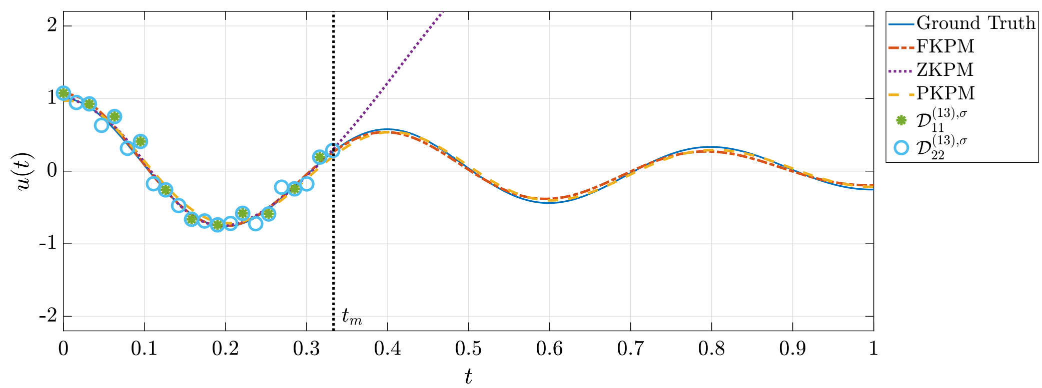

Fig. 18. Results for the Surrogation scenario using $\mathcal{D}_n^{(13),\sigma}$: ground truth, $\mathcal{D}_{11}^{(13),\sigma}$, $\mathcal{D}_{22}^{(13),\sigma}$, and the prediction of the FKPM, ZKPM, and PKPM for a single realization of the noise.

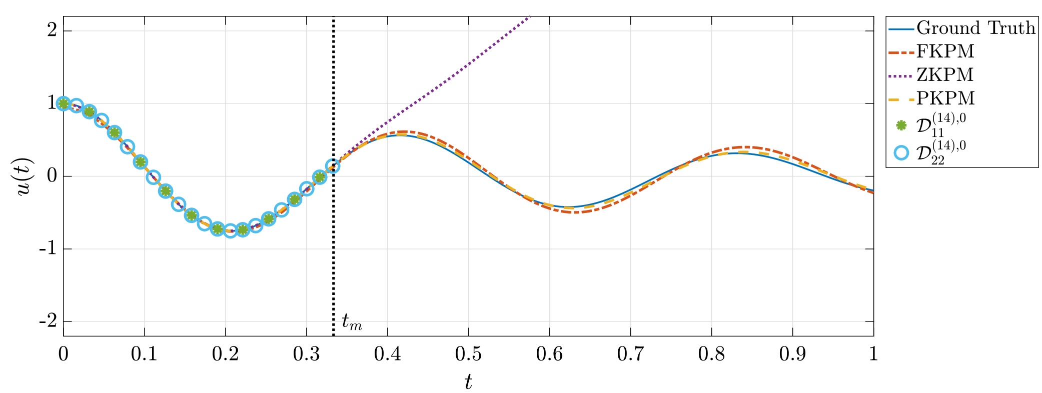

Fig. 19. Results for the Modeling scenario using $\mathcal{D}_n^{(14),0}$: ground truth, $\mathcal{D}_{11}^{(14),0}$, $\mathcal{D}_{22}^{(14),0}$, and the prediction of the FKPM, ZKPM, and PKPM.

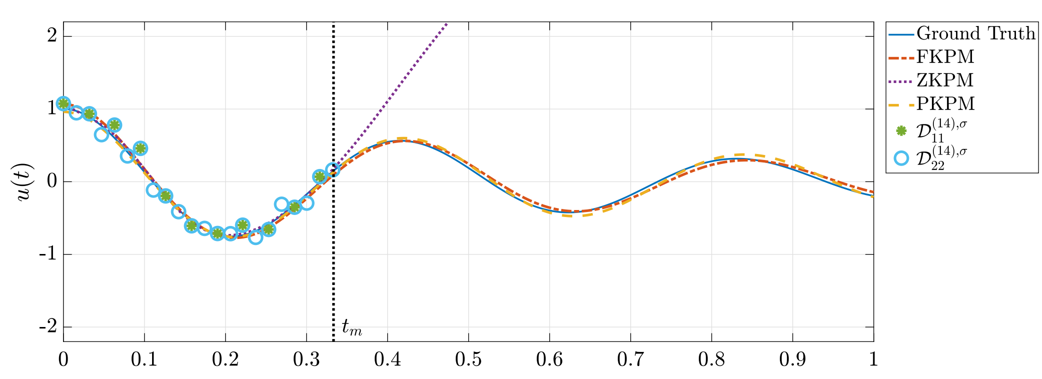

Fig. 20. Results for the Modeling scenario using $\mathcal{D}_n^{(14),\sigma}$: ground truth, $\mathcal{D}_{11}^{(14),\sigma}$, $\mathcal{D}_{22}^{(14),\sigma}$, and the prediction of the FKPM, ZKPM, and PKPM for a single realization of the noise.

Table 4
Results for the Surrogation scenario using $\mathcal{D}_n^{(13),\sigma}$: mean and standard deviation (of the 30 repetitions of the experiment) of training time, prediction time, interpolation error, and extrapolation error of FKPM (with $n = 11$), ZKPM, and PKPM (with $n = 22$).

		FKPM	ZKPM	PKPM
		($n = 11$)	($n = 22$)	
MAE	Interpolation	0.0394 ± 0.0165	0.0390 ± 0.0051	0.0343 ± 0.0091
	Extrapolation	0.0662 ± 0.0487	3.1649 ± 0.358	0.0287 ± 0.0038
Time	Training	∼ 160 s	∼ 5 s	∼ 170 s
	Predicting	∼ 10^{-3} ms	∼ 10^{-4} ms	∼ 10^{-4} ms

Table 5
Results for the Modeling scenario using $\mathcal{D}_n^{(14),0}$: training time, prediction time, interpolation error, and extrapolation error of FKPM (with $n = 11$), ZKPM, and PKPM (with $n = 22$).

		FKPM	ZKPM	PKPM
		($n = 11$)	($n = 22$)	
MAE	Interpolation	0.0233	0.0020	0.0012
	Extrapolation	0.0528	2.8706	0.0233
Time	Training	∼ 160 s	∼ 5 s	∼ 330 s
	Predicting	∼ 10^{-3} ms	∼ 10^{-4} ms	∼ 10^{-4} ms

Table 6
Results for the Modeling scenario using $\mathcal{D}_n^{(14),\sigma}$: mean and standard deviation (of the 30 repetitions of the experiment) of training time, prediction time, interpolation error, and extrapolation error of FKPM (with $n = 11$), ZKPM, and PKPM (with $n = 22$).

		FKPM	ZKPM	PKPM
		($n = 11$)	($n = 22$)	
MAE	Interpolation	0.0556 ± 0.0039	$0.0403 \pm 0.0.0060$	0.0391 ± 0.0108
	Extrapolation	0.1045 ± 0.0175	3.0229 ± 0.3624	0.0962 ± 0.0257
Time	Training	∼ 160 s	∼ 5 s	∼ 330 s
	Predicting	∼ 10^{-3} ms	∼ 10^{-4} ms	∼ 10^{-4} ms

Scopus[2.1]. The time window exploited is the same as Section 2, namely papers published after 2013. For the FKPMs and the ZKPMs, which belong to a more established field of research, we noticed that many reviews already exist that exhaustively cover all the main achievements in these industrial domains. Hence, we simply reference FKPMs and ZKPMs review papers for the reader to have an entry point for deepening the concepts. For the PKPMs, which are the main subject of our work, we reference all the most recent papers that are not covered by any review. In fact, in this case, this field of research is quite immature, and new important results are reached frequently in the last years. Because of the number of works retrieved in Scopus[2.1] with the keyword-based search we needed to filter the works based on their importance and impact so, as in Section 2, we also introduced a minimum number of citations required for the paper to be reported in this work.

In each of one Sections 5.1–5.6 a summary table, i.e., Tables 7–12 respectively, for the most relevant research paper has been reported containing: the work reference, the year of publication, the type of PKPM (i.e., Pre-, In-, and Post-Processing), the application domain, the available and/or generated data, the available prior-knowledge and/or FKPMs, the ZKPMs exploited, and a summary of the obtained results.

5.1. Extraction

The extractive industry provides raw materials and fuels essential for various industrial processes [155]. It encompasses mining, drilling, and refining operations crucial for producing metals, minerals, and energy sources like oil and gas [156]. These materials are the backbone of modern industrial production, powering manufacturing and infrastructure development worldwide [155]. Improving the processes and balancing the need for raw materials with sustainability concerns is vital for ensuring the long-term viability of both industry and the planet [156]. Predictive models, such as FKPM, ZKPM, and PKPM are increasingly employed to optimize operations, anticipate resource availability, predict the composition of the subsurface, and enhance safety protocols [11].

Numerical simulation is one of the most widely adopted FKPM. Previous works review the application of FKPM in the mining process [157], while others are devoted to related fields such as the mineral processing [158] or the tunneling [159].

The applications of ZKPMs are covered extensively in the literature. Some important works are [160,161] devoted to the oil and gas and the mineral industries respectively.

For what concerns PKPMs, some works marginally cover their application in the extractive industry [66], but a review of the topic is practically absent. We intend to fill this gap and summarize the trending applications of PKPMs in this field.

For the extraction industry, we used a particular search string[7] containing a series of keywords related to the extractive industry and the PKPMs. Since the results of this search reported more than 1400 papers we reduced this number by filtering just the papers cited (according to Scopus) by ≥25 papers for works published from 2013 and 2021, by ≥15 papers for works published in 2022, by ≥10 papers for works published in 2023, and by ≥1 papers for works published in 2024.

[7] This is the exact Scopus search string: TITLE-ABS-KEY((({theory-guided machine learning}) OR ({informed machine learning}) OR ({physics-informed machine learning}) OR ({physics-infused machine learning}) OR ({physics-guided machine learning}) OR ({physics-driven machine learning}) OR ({theory-guided data-driven}) OR ({physics-informed data-driven}) OR ({physics-infused data-driven}) OR ({physics-guided data-driven}) OR ({physics-based data-driven}) OR ({theory-guided neural network}) OR ({theory-guided neural networks}) OR ({physics-informed neural network}) OR ({physics-informed neural networks}) OR ({physics-infused neural network}) OR ({physics-infused neural networks}) OR ({physics-guided neural network}) OR ({physics-guided neural networks}) OR ({physics-driven neural network}) OR ({physics-driven neural networks}) OR (gr?y-box) OR ({hybrid data-driven}) OR ({hybrid modeling}) OR ({hybrid modelling}) OR ({hybrid machine learning}) OR ({hybrid data driven})) AND (mine OR extraction OR quarrying OR cave OR quarry OR soil OR petroleum OR petrol OR oil OR deposits OR petrochemicals OR petchems OR mineral OR fossil OR gas OR coal OR extractivism OR reservoir)) AND PUBYEAR>2012

The literature analysis shows that fossil fuel extraction is the prevalent application of PKPMs in the mining industry. In particular, great attention is reserved to the subsurface porosity and the reservoir models. Full-knowledge numerical simulations represent the traditional approach to these problems and many tools are present in the market [162]. However, they are computationally inefficient and practitioners need more viable techniques, like PKPMs [162]. The most popular PKPMs for reservoir modeling are represented by graph models derived from the domain knowledge, in which the parameters are estimated by a data-driven algorithm leveraging on historical samples [162–167]. This class of methods was tested in several fields, reaching an accuracy comparable to commercial numerical simulators while guaranteeing a lower prediction time by two orders of magnitude [167].

The reduction of the computational burden is also faced by other approaches that use PKPMs as efficient surrogates, taking advantage of neural operators, physics-informed neural networks, or outputs of an FKPM to train the model [168–171].

Reviewed works have been summarized in Table 7.

5.2. Chemical

The chemical industry is a broad sector that encompasses the production and manufacturing of a wide range of chemical substances and materials [173]. This industry typically refers to the transformation of raw materials, such as oil, air, water, metals, and minerals, into tens of thousands of different products [173]. The main subdomains within this vast industry include agrochemicals, pharmaceuticals, polymers, and oleochemicals, each serving different market needs and applications [173]. One of the primary open challenges facing the industry is sustainability, including the reduction of carbon footprint, management of chemical waste, and the adaptation to renewable resources amidst growing environmental concerns [174]. The future direction of the chemical industry is increasingly oriented toward green chemistry and biotechnology, aiming to develop processes and products that are environmentally friendly and sustainable [174]. Predictive models play a crucial role in this sector, offering profound benefits in optimizing production processes, forecasting market demands, enhancing material properties, and innovating new products [173]. By leveraging FKPMs, ZKPMs, and PKPM it is possible to reduce costs, improve efficiency, and accelerate research and development, underscoring their importance in driving the industry forward [40].

According to [40], the FKPMs currently used for the chemical industry were first introduced in 1960 by [175]. Detailed reviews on this subject can be found in [176–178].

For what concerns the ZKPMs for the chemical industry, many exhaustive reviews already exist, see [179–184] and reference therein.

For what concerns the applications of the PKPMs to the chemical industry several detailed review articles already exist [28,40,63,66,76] but they should be integrated by several relevant papers not included or published after their publication date. This work fills the gap, including papers published after 2020 that have not already been covered, to avoid overlaps with previous works.

For the chemical industry, we exploited a particular search string[8] that considers a series of keywords related to the chemical industry and the predictive models. Since the results of this search reported more than 1000 papers, we reduced this number by filtering just the papers cited (according to Scopus) by ≥15 papers for works published from 2020 and 2022, by ≥10 papers for works published in 2023, and by ≥1 papers for works published in 2024.

Process modeling is the prevalent application field of PKPMs in the chemical industry. Applications to processes span a wide range of domains, from the production of chemicals, materials, and drugs (e.g., methane in [185], polymers in [186], or β-carotene in [187]) to the treatment of wastewater [188,189]. A second field of interest in the chemical sector is the prediction of properties of substances [147,190, 191].

The most adopted type of PKPMs is the combination of an FKPM with a separated ZKPM. In particular, in several papers the output of a ZKPM is post-processed by an FKPM [187,192–194], in others the input of a ZKPM is pre-processed using an FKPM [188] or the outputs of a ZKPM and an FKPM are combined to get the final output [186]. More recent works adopt a more integrated way of combining FKPMs and ZKPMs by modifying the model inner structure [189] or the loss function [185,189]. Specifically, [185] used a physics-informed neural network to surrogate the solution of a differential equation.

Other techniques, such as in physics-informed feature-engineering, are used by [147] to predict the properties of new shape memory alloys. [190] predicted the temperature-dependence of viscosity for oxide glass-forming liquids using a zero-knowledge neural network to estimate the parameters of the full-knowledge MYEGA viscosity equation. Moreover, [191] introduced a multi-fidelity neural network (MFNN). It is composed of a “low-fidelity” neural network (NN_L), trained with the output of an FKPM, and a “high-fidelity” neural network (NN_H) that takes as additional input the output of NN_L, and is trained with experimental data.

These works in the field of PKPMs have been summarized in Table 8.

5.3. Manufacturing

The manufacturing industry encompasses the process of converting raw materials or components into finished goods through various production methods [226]. This sector involves a wide range of activities, including design, machining, fabrication, and assembly, across all industrial fields [226]. Challenges within manufacturing include improving product design, optimizing production efficiency, ensuring product quality, and navigating supply chain disruptions [227]. Predictive modeling plays a crucial role in addressing these challenges by leveraging physical knowledge and data analytics to improve processes, forecast demand, optimize inventory management, and enhance operational processes, ultimately driving efficiency and competitiveness in the industry [227]. In the new framework of smart manufacturing, the requests for superior quality, higher customization, lead time reduction, and compliance with safety it is essential to increase the performance of predictive models [35].

The class of FKPMs is mainly represented by numerical simulations [228]. In literature, it is possible to find general reviews [228] and reviews dedicated to specific fields, such as additive manufacturing [229].

For what concerns ZKPMs, an increasing amount of reviews concerning manufacturing has been published in recent years [230–232].

We also found previous reviews on PKPMs, focused on the paradigm of smart manufacturing [27,74] and additive manufacturing [82]. However, the literature lacks a work generally focused on the manufacturing

[8] This is the exact Scopus search string: TITLE-ABS-KEY((({theory-guided machine learning}) OR ({informed machine learning}) OR ({physics-informed machine learning}) OR ({physics-infused machine learning}) OR ({physics-guided machine learning}) OR ({physics-driven machine learning}) OR ({theory-guided data-driven}) OR ({physics-informed data-driven}) OR ({physics-infused data-driven}) OR ({physics-guided data-driven}) OR ({physics-based data-driven}) OR ({theory-guided neural network}) OR ({theory-guided neural networks}) OR ({physics-informed neural network}) OR ({physics-informed neural networks}) OR ({physics-infused neural network}) OR ({physics-infused neural networks}) OR ({physics-guided neural network}) OR ({physics-guided neural networks}) OR ({physics-driven neural network}) OR ({physics-driven neural networks}) OR (gr?y-box) OR ({hybrid data-driven}) OR ({hybrid modeling}) OR ({hybrid modelling}) OR ({hybrid machine learning}) OR ({hybrid data driven})) AND (chemistry OR chemical OR pharma OR pharmaceutical OR compound OR molecule OR reaction OR reactor OR culture OR bioreactor OR batch)) AND PUBYEAR>2020

Table 7
Papers about PKPMs in extraction industry.

Paper	Year	PKPM	Application domain	Available and/or generated data	Prior Knowledge and/or FKPM	ZKPM	Results & comparison between FKPM, ZKPM, and PKPM
[163]	2016	In (parameters tuning)	Reservoir waterflooding	(i) Synthetically generated data (1050 days simulated using Eclipse 100). (ii) Experimental data (5800 samples)	1-dimensional connective flow units model	Interior-point method	(i)With PKPM, NPV increased more than 3 times. PKPM is comparable to FKPM but more efficient (ii) With PKPM, +20% oil production rate and −4% water cut. No comparisons vs ZKPM
[162]	2016	In (parameters tuning)	Reservoir waterflooding	Field data (1925 days)	1-dimensional connective flow units model	Interior-point method	Graphs show that the PKPM can have an accuracy similar to FKPM simulations, but with superior efficiency. No comparisons vs ZKPM
[164]	2017	In (parameters tuning)	Reservoir waterflooding	Synthetically generated data (field example trained with simulations of 800 days)	1-dimensional connective flow units model	Ensemble smoother with multiple data assimilation	Graphs show that the PKPM can have an accuracy similar to FKPM simulations, but with superior efficiency. PKPM outperforms a previous capacitance-resistance PKPM. No comparisons vs ZKPM
[165]	2018	In (parameters tuning)	Reservoir waterflooding	Synthetically generated data (field example trained with Eclipse 100 simulations of 2050 days)	1-dimensional connective flow units model	Ensemble smoother with multiple data assimilation	In the field test, optimization with PKPM increases NPV of 40%. No comparisons vs FKPM and ZKPM
[166]	2018	In (parameters tuning)	Unconventional reservoir flow modeling	Field data (300 days)	Diffusive diagnostic function	Ensemble smoother with multiple data assimilation	Graphs show that the PKPM can have a good accuracy on the testset
[172]	2020	Post	Hydraulic fracturing	Synthetically generated data (2841 snapshot of 251 points each)	Fracture propagation PKN model	Neural network	Graphs shows that PKPM and ZKPM perform similarly in interpolation. PKPM is much better in extrapolation. FKPM can predict the trend, but with low accuracy.
[168]	2020	Pre & In (surrogate)	Reservoir simulation	Synthetically generated data (5700 data points)	AD-GPRS full-order simulations and Darcy's model	Neural network	Tested in 100 cases, PKPM can predict the production with an average error of 0.14 and is 1000 times faster than the FKPM simulation. No comparisons vs ZKPM
[53]	2021	Pre	Carbonate rock permeability estimation	Experimental data (1100 samples of 100 × 100 × 160 voxels) and data from literature (400 samples). Ground-truth porosity computed using Lattice Boltzmann method	Pore network model	Linear regression, Support vector regression, Gradient boosting, Random forest, Deep neural network, Convolutional neural network	PKPMs tested varying the architecture. PKPMs give a best value of R^2 of 0.87, FKPM 0.15. No comparisons vs ZKPM
[169]	2021	Pre (surrogate)	Reservoir pressure management	Synthetically generated data (variable number of samples)	Theis model	Neural network	PKPM can predict overpressure with RMSE < 0.1 mmH_2O. No comparisons vs FKPM and ZKPM.

(continued on next page)

industry, which we intend to cover.

For what concerns the manufacturing industry, we used a particular search string[9] considering a series of keywords related to the manufacturing industry and the predictive models. The results of this search reported more than 2900 papers, and then we reduced this number by filtering just the papers cited (according to Scopus) by ≥50 papers

[9] This is the exact Scopus search string: TITLE-ABS-KEY(((({theory-guided machine learning}) OR ({informed machine learning}) OR ({physics-informed machine learning}) OR ({physics-infused machine learning}) OR ({physics-guided machine learning}) OR ({physics-driven machine learning}) OR ({theory-guided data-driven}) OR ({physics-informed data-driven}) OR ({physics-infused data-driven}) OR ({physics-guided data-driven}) OR ({physics-based data-driven}) OR ({theory-guided neural network}) OR ({theory-guided neural networks}) OR ({physics-informed neural network}) OR ({physics-informed neural networks}) OR ({physics-infused neural network}) OR ({physics-infused neural networks}) OR ({physics-guided neural network}) OR ({physics-guided neural networks}) OR ({physics-driven neural network}) OR ({physics-driven neural networks}) OR (gr?y-box) OR ({hybrid data-driven}) OR ({hybrid modeling}) OR ({hybrid modelling}) OR ({hybrid machine learning}) OR ({hybrid data driven})) AND (manufacturing OR manufacture OR additive OR production OR design OR machined OR machining OR milling OR product OR metal OR plastic OR composite OR wood OR ceramic OR cnc)) AND PUBYEAR>2012

Table 7 (*continued*).

[170]	2022	In (surrogate)	Porous media gas drainage	Synthetically generated data (11 saturation profiles made of 100 points each)	Buckley–Leverett model	Neural network	The best PKPM gives a $MEAresidual$ of 0.01. ZKPM neural network is comparable in interpolation, but accuracy drops in extrapolation. No comparisons vs FKPM
[171]	2022	In (surrogate)	Subsurface oil/water two-phase flow modeling	Synthetically generated data (1000 samples simulated with SGeMS's sequential Gaussian simulation)	two-phase underground oil/water flow 2D model	Fourier neural operator network	Tested in reservoir saturation prediction the proposed PKPM reduces $RMSE$ by more than 46% with respect previous PKPMs. No comparisons vs FKPM and ZKPM
[167]	2023	In (parameters tuning)	Reservoir waterflooding	Synthetically generated data (field example trained with Eclipse 100 simulations of 2400 days)	1-dimensional connective flow units model	Ensemble smoother with multiple data assimilation	Results of PKPM are comparable to FKPM (Eclipse 100) and it is 2 order of magnitude more efficient. No comparisons vs ZKPM.

for works published from 2013 and 2021, by ≥30 papers for works published in 2022, by ≥10 papers for works published in 2023, and by ≥1 papers for works published in 2024.

The results show how PKPMs have been implemented in many fields, such as design [208,209], quality prediction [196], quality control [198,203], treatments [200], machining processes [204], and performance prediction [206].

Modern technologies exploit PKPMs more than the traditional ones. For example, in additive manufacturing PKPMs have been leveraged for several scenarios, from melting pool modeling [199] to defect prediction [51,196] and fatigue life prediction [205,210,211]. It is also worth noting the works of [197,200] which applied PKPMs in the world of composite materials.

Several typologies of PKPM have been exploited. Physics-informed neural network are the most adopted [198,200–203,206,208–210], but other techniques are present such as feature pre-processing [51] and surrogate models of computationally expensive FKPMs [196, 197].

From these works, we can infer that PKPMs in manufacturing are mainly used to improve computational efficiency and tackle the traditional models' data dependency. However, it is worth highlighting also the use of PKPMs to identify the input parameters (i.e., inverse problem) [203] and to augment the generalization ability of traditional ZKPMs [51].

Table 9 summarizes the reviewed papers.

5.4. Transportation

The transportation industry is the backbone of the world's industry, involving the movement of goods and passengers [233]. The most important branches are automotive, marine, aerospace, and railway, but the transportation industry also involves traffic management and the logistics industry [234]. Predictive models are indispensable tools to forecast trends, optimize operations, improve sustainability, and mitigate risks [235]. In the automotive sector, predictive modeling aids in demand forecasting, maintenance schedules, and autonomous vehicle development, revolutionizing how we commute and transport goods [236–238]. Similarly, in the marine industry, predictive models optimize vessel routing, predict weather patterns, and enhance navigation safety, ensuring seamless maritime and port operations [239]. In aerospace, predictive modeling plays a crucial role in aircraft design, operations optimization, and maintenance scheduling, advancing air travel toward greater reliability and fuel efficiency [240]. Finally, in the rail industry, predictive modeling facilitates efficient scheduling, maintenance planning, and infrastructure management, contributing to the reliability and sustainability of rail transportation networks [241].

A general review of FKPMs applied to transportation can be found in [242].

A plethora of surveys cover the applications of ZKPMs in the transportation industry. Some notable examples are [34,243], devoted to goods transportation and traffic management, respectively.

The work on PKPMs in the transportation industry is still limited. Some review papers cover specific fields, such as fuel consumption [65, 80] or ship motion [244]. However, to the best of our knowledge, no works cover the entire sector. To bridge this gap, we searched related papers using a specific string[10] to include all the inherent papers. Since the results of this search reported more than 1200 papers, we reduced this number by filtering just the papers cited (according to Scopus) by ≥25 papers for works published from 2013 and 2021, by ≥15 papers for works published in 2022, by ≥10 papers for works published in 2023, and by ≥1 papers for works published in 2024.

Results show how ship operation optimization is one of the most explored fields in the context of PKPMs for transportation. Several works apply PKPMs to predict and reduce fuel consumption [212,215,216], to model the ship behavior [213,214,220,225], and to consolidate the use of recent forms of propulsion [224]. PKPMs have also been implemented to model the behavior of batteries in relation to the operation of the vehicles that they supply [221,222]. The study of the cumulative damage of aircraft is another application of PKPMs [217,223].

Popular PKPMs are physics-inspired structures with data-driven parameter estimations [213,214,216] and physics-inspired regularization [107,212] and architectures [217,222].

In most cases, PKPMs are used to improve the accuracy of traditional methods, although it is noteworthy the use of PKPMs to impose a knowledge-based constraint [219] and reduce computational time [218].

Table 10 summarizes the reviewed papers.

[10] This is the exact Scopus search string: TITLE-ABS-KEY((({theory-guided machine learning}) OR ({informed machine learning}) OR ({physics-informed machine learning}) OR ({physics-infused machine learning}) OR ({physics-guided machine learning}) OR ({physics-driven machine learning}) OR ({theory-guided data-driven}) OR ({physics-informed data-driven}) OR ({physics-infused data-driven}) OR ({physics-guided data-driven}) OR ({physics-based data-driven}) OR ({theory-guided neural network}) OR ({theory-guided neural networks}) OR ({physics-informed neural network}) OR ({physics-informed neural networks}) OR ({physics-infused neural network}) OR ({physics-infused neural networks}) OR ({physics-guided neural network}) OR ({physics-guided neural networks}) OR ({physics-driven neural network}) OR ({physics-driven neural networks}) OR (gr?y-box) OR ({hybrid data-driven}) OR ({hybrid modeling}) OR ({hybrid modelling}) OR ({hybrid machine learning}) OR ({hybrid data driven})) AND (ship OR aerospace OR plane OR wing OR car OR automotive OR traffic OR naval OR freight OR airlines OR marine OR rail OR infrastructures OR railway OR aircraft OR transportation OR vehicle)) AND PUBYEAR>2012

Table 8
Papers about PKPMs in chemical industry.

Paper	Year	PKPM	Application domain	Available and/or generated data	Prior Knowledge and/or FKPM	ZKPM	Results & comparison between FKPM, ZKPM, and PKPM
[147]	2021	Pre	Shape memory alloys design	Data of existing alloys (528 from literature and 26 from authors' lab)	Thermodynamics and kinetics of phase transformations	Gaussian process	PKPM is more accurate than ZKPM in predicting $\tilde{T}$ and T. No comparisons vs FKPM
[190]	2021	Post	Viscosity prediction	Data extracted from SciGlass database (17584 data of 847 oxide liquids)	MYEGA viscosity equation	Neural network	PKPM predicts testset with $R^2 = 0.97$. No comparisons vs FKPM and ZKPM
[186]	2021	Post	Semi-batch polymerization reactors	Synthetically generated (125 batches)	Physical equations (e.g., mass and energy balances)	Symbolic regression	PKPM is qualitatively more accurate than the simplified FKPM. No comparisons vs ZKPM
[192]	2021	Post	Chromatographic processes	Synthetically generated data (17 BT curves) and experimental data (17 BT curves)	Lumped kinetic model	Neural network	PKPM 3 times more accurate than FKPM in interpolation and extrapolation. No comparisons vs ZKPM
[193]	2021	Post	Czochralski silicon single crystal growth process	Experimental data (5000 sets of crystal growth, continuously selected at intervals of 5 data points)	Hydrodynamic and geometric model	LSTM network	Compared 3 different LSTMs inside PKPM. M-LSTM-HW is the most accurate in all the tests. No comparisons vs FKPM and ZKPM
[187]	2021	Post	Yeast astaxanthin production	Experimental data (quantity not specified)	Kinetic model	Gaussian process	PKPM has higher accuracy and lower uncertainty than FKPM, for all the predicted quantities. No comparisons vs ZKPM
[185]	2021	In (surrogate)	Catalytic CO_2 methanation	Initial and boundary conditions and synthetically generated data (Inverse problem. Different sized datasets)	Governing equations	Neural network	Once trained, PKPM is more efficient than FKPM (numerical solution). Error $< 0.3\%$ in inverse problem (parameters identification).
[194]	2022	Post	β-carotene production	Experimental data (quantity not specified)	Kinetic model	Neural network	PKPM is more accurate than FKPM, for all the predicted quantities. No comparisons vs ZKPM
[188]	2022	Pre	Wastewater treatment	Data from a full-scale WWTP (1248 data after pre-treatment)	Activated sludge model	LSTM network	PKPM is more accurate than ZKPM (MSE $-22,5\%$) and FKPM (MSE $-93,0\%$)
[195]	2022	In & Post	Industrial fermentation process	Experimental data (from a sponsoring industry, quantity not specified)	Kinetic model	Neural network	PKPM is more accurate than FKPM, for all the predicted quantities. No comparisons vs ZKPM
[191]	2023	Pre	Nanofluids viscosity prediction	Experimental data (1425 data of 19 class of nanofluids)	Viscosity theoretical models (Einstein, Batchelor, Brinkman, Masoumi, and Udawattha)	Neural network	PKPM is more accurate than ZKPM ($R^2 = 0.991$ vs $R^2 = 0.977$). No comparisons vs FKPM
[189]	2023	Pre & In	(i) Semi-batch reactor. (ii) Wastewater treatment	(i) Synthetically generated data (500 data from numerical solution). (ii) Data from literature (1345 data)	Chemical equations	Neural network and Recurrent neural network	Tested several structures of PKPM and compared with ZKPMs. PKPMs outperform ZKPMs in all the tests. No comparisons vs FKPM

Table 9
Papers about PKPMs in manufacturing industry.

Paper	Year	PKPM	Application domain	Available and/or generated data	Prior Knowledge and/or FKPM	ZKPM	Results & comparison between FKPM, ZKPM, and PKPM
[196]	2020	Pre (surrogate)	Additive manufacturing defect prediction	Experimental (320 hatches) and synthetically generated data (3200 simulation blocks)	Numerical simulation of plastic damage model	Neural network	In the LPBF case, the $F - Score$ for bulk and overhang is 87.5% and 79.6% for PKPM and ZKPM respectively. No comparisons vs FKPM.
[197]	2020	Pre (surrogate)	Composite materials damage characterization	Synthetically generated (10000 simulations of LS-DYNA) and experimental data (3 ICT tests)	Numerical simulation of plastic damage model	Neural network	The $RMSE$ of PKPM with respect to FKPM is $1.9\ \mathrm{kJ}\frac{\mathrm{kJ}}{\mathrm{m}^2}$ for fracture energy. No comparisons vs ZKPM.
[198]	2020	In (surrogate)	Surface cracks ultrasound quantification	Experimental data (240 × 240 grid, 1024 time instants, 3 different wave-crack orientations)	Acoustic wave equation	Neural network	Inverse problem. With 20% of data PKPM can predict sound speed with an error of 1%. No comparisons vs FKPM and ZKPM
[199]	2021	Pre & In	Additive manufacturing melt pool modeling	Synthetically generated data (quantity not specified)	Momentum, mass and energy conservation	Neural network	PKPM results are comparable to numerical FKPMs. No comparisons vs ZKPM
[51]	2021	Pre	Additive manufactured components porosity analysis	Experimental data (549 3D-printed components)	Physical parameters affecting the process	Linear regression, Gaussian process, Support vector regression	Accuracy of PKPMs and ZKPMs are comparable, but PKPMs are independent from the specific machine parameters. No comparisons vs FKPM
[200]	2021	In (surrogate)	Composite materials curing process	Initial and boundary conditions	Thermodynamics equations	Neural network	Max error of PKPM with respect to FKPM is $0.97\ °C$. PKPM can extrapolate better than ZKPM
[201]	2021	In (surrogate)	Component elastodynamics	Initial and boundary conditions	Elastodynamics equations	Neural network	Qualitative comparison of PKPMs with different hyperparameters and reference FKPM numerical solution. No comparisons vs ZKPM
[202]	2021	Pre & In (surrogate)	Composite materials curing process	Initial and boundary conditions	Heat transfer law and degree of cure equation	Neural network	PKPM compared to the FKPM numerical simulation. PKPM 10 times faster and error $< 1.6K$ and 0.007 of temperature and degree of cure respectively. No comparisons vs ZKPM
[203]	2022	Pre & In (surrogate)	Internal structure and defects of components	Different data for each case (initial and boundary conditions, internal samples)	Mechanical laws (different assumptions for different cases)	Neural network	Inverse problem. PKPM predicts the defect parameters with a max error of 6.8%. No comparisons vs FKPM and ZKPM.
[204]	2022	Pre & In	Machining tool wear	Experimental data (3 datasets of 315 data)	Tool wear empirical equation	Neural network	PKPM predicts the wear with an $RMSE$ of 3.17, 3.01, and 7.58 in x, y, and z respectively. PKPM is better than all the tested ZKPMs, no comparisons vs FKPM
[205]	2022	Pre & In	Additive manufactured components fatigue	Data from literature (12 samples)	Fracture mechanics model	Neural network	In the test, PKPM and ZKPM give $R^2 = 0.591$ and $R^2 = 0.322$ respectively. No comparisons vs FKPM
[206]	2023	In	Fatigue life prediction	Data from literature (3 materials: WIRE 85 samples, 2024-T4 252 samples, AAW 200 samples)	Fatigue model	Neural network	Measure of the physical consistency: PKPM and ZKPM has a minimum value of -0.99 and -0.53. No comparisons vs FKPM

(continued on next page)

Table 9 (*continued*).

[207]	2023	In (surrogate)	Finite-strain plasticity modeling	Initial and boundary conditions	Finite-strain elastoplasticity model	Neural network	3 step loading test: PKPM gives an error of 3.58% with respect to the FKPM. No comparisons vs ZKPM
[208]	2023	In (surrogate)	Topological optimization	Boundary conditions	Deep energy method	Neural network	Images show a good agreement between PKPM and FKPM. PKPM is less sensitive to the number of nodes
[209]	2023	In (surrogate)	Topological optimization	Boundary conditions	Total potential energy model	Neural network	In the 3D example the difference between PKPM and FKPM is less than 5%. No comparisons vs ZKPM
[210]	2023	Pre & In	Additive manufactured components fatigue	Data from literature (561)	Fracture mechanics model	Neural network	PKPM more interpretable than ZKPM. PKPM gives an MSE 20% lower than ZKPM. No comparisons vs FKPM
[211]	2023	(i) Pre (ii) Post	Additive manufactured components fatigue	Experimental data (62 tests)	Paris' law, $\delta\sigma$ model	Neural network, Support vector regression	Several PKPM architectures tested. $RMSE$ given by all PKPMs is less than FKPMs and ZKPMs

5.5. Energy

The energy industry is a fundamental pillar supporting the functioning of societies, providing essential power for cities and industries [245]. Renewable energy sources such as solar, wind, and hydroelectric power are increasingly gaining prominence due to their sustainability and environmental benefits, while traditional fossil fuels continue to supply a significant portion of global energy needs [246]. Energy storage methods, including batteries, pumped hydro storage, and emerging technologies like hydrogen fuel cells, are critical for balancing supply and demand, ensuring stability in power grids, and facilitating the integration of intermittent renewable energy sources [247]. Predictive modeling techniques are increasingly applied in the energy sector to forecast energy production, consumption, and storage, optimize components' maintenance, enhancing operational efficiency [245].

In the literature, we found several reviews on using FKPMs in the energy industry. We highlight [248] focused on thermal plants, [249] focused on the application of FKPMs and ZKPMs for the solar energy forecast, and [250] focused on the hydrogen fuel cells.

The applications of ZKPMs in smart energy management are reviewed in [245]. We also cite the works of [251,252], devoted to two more specific fields, such as renewables and batteries.

For what concerns PKPMs, an increasing interest is rising toward batteries modeling, reviewed in [73,82]. [66] reviewed the serial and parallel combination of FKPMs and ZKPMs in different sectors, including the energy industry. Finally, [77] cover the applications of PKPMs for power systems.

Our work aims to provide a comprehensive overview of applications in the energy industry. In order to do that, we selected a search string[11] considering several related keywords. Since the results of this search reported more than 2900 papers, we reduced this number by filtering just the papers cited (according to Scopus) by ≥30 papers for works published from 2013 and 2021, by ≥20 papers for works published in 2022, by ≥10 papers for works published in 2023, and by ≥1 papers for works published in 2024.

Following the general trend of industry [73], results highlight a significant interest in batteries and renewables. Many works address the battery modeling, in particular, to estimate the states of charge and health during operation [253–257]. Furthermore, [258,259] apply a PKPM to optimize wind turbine maintenance. The work of [260] tries to forecast photovoltaic production. Other notable works devoted to relevant sectors are [261,262], inherent to fuel cell operation optimization and nuclear reactor modeling, respectively.

The most common type of PKPM in the examined literature is the feeding of ZKPMs with additional features outputted by FKPMs [255–257,260,263,264]. The second most adopted PKPM is the surrogation of heavy computational FKPMs with more efficient PKPMs [254,261,262].

Table 11 summarizes the reviewed papers.

5.6. Construction

The construction industry includes a diverse range of activities, including residential, commercial, and infrastructure development [267]. With increasing concerns about sustainability and environmental impact, there is a growing emphasis on energy efficiency within construction practices [268]. Efforts to enhance energy efficiency in buildings have become of paramount importance in the construction industry, driven by both environmental concerns and regulatory requirements [268]. Incorporating technologies like smart insulation, efficient lighting, and renewable energy sources can significantly reduce energy consumption and carbon emissions in buildings [269]. Leveraging predictive modeling and data analytics holds promise in optimizing resource utilization, enhancing project planning, and mitigating risks [270]. FKPMs, ZKPMs, and PKPMs aid in forecasting project outcomes, identifying potential challenges, and optimizing building efficiency, ultimately contributing to a more sustainable and cost-effective building life cycle.

An extensive review of the applications of FKPMs in the building sector was proposed by [271]. Another interesting work is [272], concerning structural optimization in civil engineering.

The work of [273] reviews the applications of ZKPMs during the whole life cycle of the buildings. An interesting survey of ZKPMs applications to structural health monitoring was proposed by [274].

[11] This is the exact Scopus search string: TITLE-ABS-KEY(((({theory-guided machine learning}) OR ({informed machine learning}) OR ({physics-informed machine learning}) OR ({physics-infused machine learning}) OR ({physics-guided machine learning}) OR ({physics-driven machine learning}) OR ({theory-guided data-driven}) OR ({physics-informed data-driven}) OR ({physics-infused data-driven}) OR ({physics-guided data-driven}) OR ({physics-based data-driven}) OR ({theory-guided neural network}) OR ({theory-guided neural networks}) OR ({physics-informed neural network}) OR ({physics-informed neural networks}) OR ({physics-infused neural network}) OR ({physics-infused neural networks}) OR ({physics-guided neural network}) OR ({physics-guided neural networks}) OR ({physics-driven neural network}) OR ({physics-driven neural networks}) OR (gr?y-box) OR ({hybrid data-driven}) OR ({hybrid modeling}) OR ({hybrid modelling}) OR ({hybrid machine learning}) OR ({hybrid data driven})) AND (energy OR energies OR renewable OR wind OR turbine OR grid OR battery OR solar OR photovoltaic OR geothermal OR heat OR nuclear OR electricity OR lithium OR fuel OR cell)) AND PUBYEAR>2012

Table 10
Papers about PKPMs in transportation industry.

Paper	Year	PKPM	Application domain	Available and/or generated data	Prior Knowledge and/or FKPM	ZKPM	Results & comparison between FKPM, ZKPM, and PKPM
[212]	2015	In	Ship efficiency estimation	Performance evaluated with several different dataset sizes	Ship fuel consumption model	Kernel method	The best PKPM predicts the fuel consumption with $MAPE = 1.49\%$, FKPM with 20.95% and ZKPM (best) with 1.71%
[213]	2015	In (parameters tuning)	Ship maneuvering model	Experimental data (quantity not specified)	Dynamical model	Least-squares support vector regression	Qualitatively PKPM is comparable to ZKPM but less accurate than FKPM
[214]	2015	In (parameters tuning)	Ship maneuvering model	Synthetically generated data (graphs shows a simulation of about 400s)	Dynamical model	ϵ support vector regression	PKPM is more accurate than FKPM and ZKPM in all the tests
[215]	2017	(i) Pre (ii) In	Ship trim optimization for fuel consumption	Performance evaluated with several different sized datasets	Ship fuel consumption model	Regularized least squares, lasso regression, and random forest	Fuel consumptions $MAPE$: FKPM 20.95%, best ZKPM 1.95%, best PKPM Pre-processing 0.83%, , best PKPM In-processing 0.95%
[216]	2019	In (parameters tuning)	Ship fuel consumption forecast	Data sampled in 7 years of ship operation (daily sampling)	Fuel consumption model	Genetic algorithm	The proposed PKPM predicts the consumption with $R^2 = 0.90$ (a previous PKPM with $R^2 = 0.88$). No comparisons vs FKPM and ZKPM
[116]	2020	Pre	Propeller noise prediction	Experimental data (258 tests in the cavitation tunnel of University of Genoa)	Cavitation vortex and high-frequency noise models	Neural network	In the interpolation test MAE of PKPM is -20% and -82% with respect to ZKPM and FKPM respectively. In the extrapolation test MAE of PKPM is -29% and -69% with respect to ZKPM and FKPM respectively.
[217]	2020	Pre & In	Wings cumulative damage modeling	Synthetically generated data (15 planes, 174000 cycles each)	Paris' law	Recurrent neural network	The accuracy of PKPM on the test set is 96.55%. No comparisons vs FKPM and ZKPM
[218]	2020	In	Traffic state estimation	Synthetically generated data (1000 samples extracted from a 500 × 500 spatiotemporal grid)	Lighthill–Whitham–Richards model	Neural network	Eulerian test accuracy: PKPM 73.7% and ZKPM 37%. Lagrangian test accuracy: PKPM 79% and ZKPM 86.1%. PKPM > 2 times faster than ZKPM. No comparisons vs FKPM
[219]	2021	In	Port state control	Total of 1,974 records (Hong Kong port)	Monotonicity regarding ship flag/recognized organization/company performance	Extreme gradient boosting	PKPM improves by 20% the current scheduling method. PKPM outperforms the ZKPM XGBoost
[220]	2021	Pre	Ship maneuvering model	Synthetically generated (70 simulations of 600 samples) and experimental data (quantity not specified)	Hydrodynamic model	Neural network	In simulation PKPM predicts the trajectory with an error 90% lower than FKPM. No comparisons vs ZKPM
[221]	2021	Pre & In	Unmanned aerial vehicle batteries modeling	Data from NASA's prognostic Center of Excellence Data Repository (36 discharge curves)	Electrochemical model	Recurrent neural network	Tested on random loading cases, PKPM can predict voltage with $RMSE < 0,088\%$ for 7 batteries out of 8. No comparisons vs FKPM and ZKPM
[222]	2023	Pre	Vehicle batteries heat generation estimation	Virtual battery data for five standard driving tests	Single particle model with thermodynamics	LSTM network	Comparison of different architectures of PKPM. The best can predict heat generation with an $RMSE = 1.428$ for the WLTP standard. No comparisons vs FKPM and ZKPM

(continued on next page)

Table 10 (*continued*).

[223]	2023	Pre	Aerospace alloys fatigue modeling	Experimental data (500 samples)	Basquin model, Smith-Watson-Topper model, Neuber's rule	Support vector machine, random forest, extreme gradient boosting	PKPM improves by 2% the predictions that lie in the ±2 error bands. Graphs shows that the MSE of PKPM is about 30% lower than FKPM. No comparisons vs ZKPM
[224]	2023	Pre	Safety assessment of solid oxide fuel cell for ships	Synthetically generated and experimental data (4392 samples)	Fuel cell electrochemical model	Gradient boosting	In predicting leaks, PKPM has an accuracy 11% better than ZKPM. No comparisons vs FKPM
[225]	2024	In & Post	Ship speed prediction	Experimental data (from two ships 63000 and 8600 samples respectively)	P-v curve in calm water	Neural network, extreme gradient boosting	PKPM is 30% more accurate than ZKPM. No comparisons vs FKPM

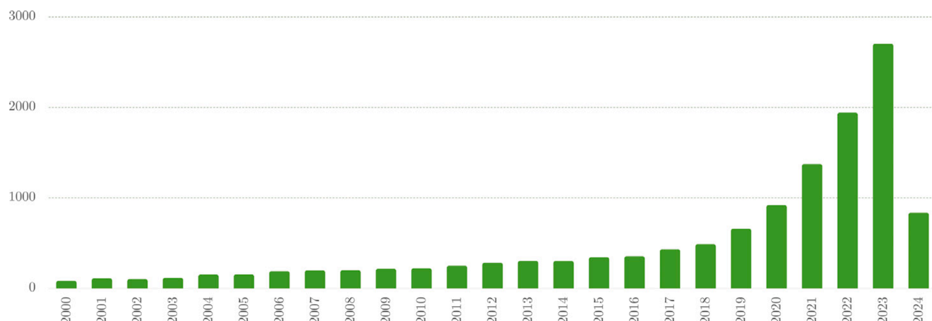

Fig. 21. Number of papers resulting from a general PKPM-related search in Scopus, grouped by year.

A large number of review papers cover the use of PKPMs in the field of building efficiency [62,64,67,69,71]. The work of [75] provides a wider view of PKPMs applied to civil engineering, focusing on building information modeling, structural health monitoring systems, and structural design and analysis. This work will give an updated overview of the PKPMs applied to the entire construction industry, excluding previously covered works. We used a specific search string[12] with the goal of covering the most representative works. The search reported more than 1800 papers, then we reduced this number by filtering just the papers cited (according to Scopus) by ≥40 papers for works published from 2013 and 2021, by ≥20 papers for works published in 2022, by ≥10 papers for works published in 2023, and by ≥1 papers for works published in 2024.

The results show that the main use of PKPMs in the construction industry is related to building efficiency, followed by structural settlement and health monitoring. Several works address the problem of building thermal modeling, to optimize the building efficiency [275–280]. Another group of works tackles the problem of structural integrity [281–283], while others address the interaction between structures and soil [284,285].

Most of the PKPMs use a physics-informed regularization term in the loss function as a method to increase the model performance [278,279,281,282,285]. Another popular class uses a physics-inspired structure with data-driven parameters tuning [275,280]. In particular, this class of methods was extensively covered in previous review works [62,64,67,69,71].

Table 12 summarizes the reviewed papers.

6. Open problems and future perspectives

While FKPMs and ZKPMs approaches are already widely adopted, PKPMs are gaining traction (see Fig. 21), despite major challenges due to their relatively recent emergence [42].

Despite the rapid progress, there remains a conspicuous shortage of professionals simultaneously trained on the specific domain and machine learning hindering the widespread adoption of PKPMs [83]. PKPMs have been developed for a limited number of applications, addressing only a fraction of the vast array of conceivable problems [42]. PKPM methods promise enhanced performance, but realizing this advantage often demands considerable development efforts [35]. Unlike traditional models, which may require a less upfront investment, PKPMs sometimes incur higher development costs, necessitating careful evaluation of the trade-offs between development expenses, computational effort, and performance improvements [35].

PKPMs' advantages are not limited to improving model accuracy, but they can be a breakthrough in several scenarios. Particularly noteworthy are cases with sparse data but rich domain knowledge, where PKPMs can effectively outperform traditional ZKPMs [41]. In certain situations, PKPMs can offer significant advantages in terms of explainability, providing insights into the underlying physical processes driving model predictions [27]. The possibility to impose knowledge-based constraints can substantially improve the models' trustworthiness [72]. Furthermore, the pre-training of models using physics-based knowledge, followed by fine-tuning with available data, has demonstrated remarkable success, ensuring an augment of the performance [78]. Finally, PKPMs shine in inverse problems. The inverse problem application has the potential to become the foremost application of PKPMs, offering significant capabilities for prescriptive analysis and optimization

[12] This is the exact Scopus search string: TITLE-ABS-KEY((({theory-guided machine learning}) OR ({informed machine learning}) OR ({physics-informed machine learning}) OR ({physics-infused machine learning}) OR ({physics-guided machine learning}) OR ({physics-driven machine learning}) OR ({theory-guided data-driven}) OR ({physics-informed data-driven}) OR ({physics-infused data-driven}) OR ({physics-guided data-driven}) OR ({physics-based data-driven}) OR ({theory-guided neural network}) OR ({theory-guided neural networks}) OR ({physics-informed neural network}) OR ({physics-informed neural networks}) OR ({physics-infused neural network}) OR ({physics-infused neural networks}) OR ({physics-guided neural network}) OR ({physics-guided neural networks}) OR ({physics-driven neural network}) OR ({physics-driven neural networks}) OR (gr?y-box) OR ({hybrid data-driven}) OR ({hybrid modeling}) OR ({hybrid modelling}) OR ({hybrid machine learning}) OR ({hybrid data driven})) AND (construction OR building OR concrete OR demolition OR bricklayers OR civil OR offshore OR residential OR plant OR bridge OR hvac)) AND PUBYEAR>2012

Table 11
Papers about PKPMs in energy industry.

Paper	Year	PKPM	Application domain	Available and/or generated data	Prior Knowledge and/or FKPM	ZKPM	Results & comparison between FKPM, ZKPM, and PKPM
[253]	2017	Post	Modeling SOC and temperature of batteries	Experimental data (PCD, HPPC e CCD tests. Quantity not specified)	Equivalent circuit and thermal models	Neural network	PKPM tested varying conditions and hyperparameters. PKPM validated for practical battery management systems. No comparisons vs FKPM and ZKPM
[265]	2020	Post	Critical heat flux	Data from literature (1865 test cases)	Liu model	Neural network, Random forest	$rRMSE$ is 5.5%, 30.9%, and 13.6% for the best PKPM, FKPM, and ZKPM respectively
[258]	2020	Pre & In	Wind turbine bearings	Data from literature (NREL database, hourly from 2007 to 2013) and synthetically generated (bearing life curves)	Bearing models	LSTM network	PKPM qualitatively has a good adherence with the bearing damage curve. No comparisons vs FKPM and ZKPM
[254]	2021	Pre (surrogate)	Battery electrode-level state estimation	Synthetically generated data (75 simulations of power profiles)	Electrochemical and thermal models	LSTM network	The maximum value of $RMSPE$ for all the quantities predicted by PKPM is 2.93% (without noise). Improvement between 75% and 95% with respect to ZKPM. No comparisons vs FKPM
[263]	2021	Pre	Prediction of polarization curves of solid oxide fuel cell anodes	Data from literature (quantity not specified)	Fundamental conservation laws of mass and electrical charge	Neural network	PKPM has a MSE that is 63% lower than FKPM. No comparisons vs ZKPM
[259]	2022	Pre & In	Wind turbine bearings	Data from literature (NREL database, hourly from 2007 to 2013) and synthetically generated (bearing life curves)	Bearing models	LSTM network	PKPM trained with different sized datasets. Best $RMSE$ for grease damage is 0.021. Qualitatively better than ZKPM. No comparisons vs FKPM
[262]	2022	Pre (surrogate)	Nuclear reactor modeling	Synthetically generated data (18480 samples)	Neutronic field simulation	Decision tree, k-nearest-neighbors	In the forward case, the PKPMs have a reconstruction error of 2.7% with respect to the FKPM simulation. PKPM is 3 orders of magnitude faster. No comparisons vs ZKPM
[255]	2022	Pre	Battery health management	Data from NASA Ames Prognostics Center of Excellence (9 different datasets)	Calendar and cycle aging models	LSTM network	PKPM more accurate than FKPM and ZKPM (Best $MAPE$: PKPM 0.92%, FKPM 3.59%, and ZKPM 4.12%)
[260]	2023	Pre	Photovoltaic power forecasting	Experimental data (from 14 plants) and numerical (weather predictions)	PV model chain	Neural network	With data sampled for more than 1 year, PKPM more accurate than FKPM. No comparisons vs ZKPM
[257]	2023	Pre	Battery health prognosis	(i) Data from NASA (first 90 life cycles of 3 batteries and first 60 life cycles of 1 battery). (ii) Experimental data (1400 life cycles)	Pseudo 2-dimensional electrochemical model	LSTM network	(i) PKPM gives $R^2 > 0.711$. (ii) PKPM gives $R^2 > 0.989$. No comparisons vs FKPM and ZKPM
[256]	2023	Pre & Post	Battery voltage modeling	(i) Synthetically generated data (DFN model). (ii) Experimental data (commercial battery). Quantity not specified	(i) Single particle model with thermal dynamics. (ii) Nonlinear double capacitor equivalent circuit model	Neural network	(i) PKPM more accurate than FKPM in all the tests. (ii) PKPM more accurate than FKPM and ZKPM in all the tests

(*continued on next page*)

Table 11 (*continued*).

[266]	2024	Pre	Power system load margin prediction	(i) Synthetically generated data (1000 samples from nominal IEEE 68-bus). (ii) Synthetically generated data (1000 samples from nominal BIPS)	Load margin assessment model	Neural network	PKPM more accurate and robust than ZKPM. PKPM is slower than ZKPM in training. Similar in prediction times. No comparisons vs FKPM
[261]	2024	Pre (surrogate)	Solid oxide fuel cells operating condition optimization	Synthetically generated data (50000 simulated data)	Local overpotential model	Neural network	PKPM 10000 times faster than FKPM. Error 0.513% and 2.86% with respect to FKPM and experiments, respectively. No comparisons vs ZKPM
[264]	2024	Pre	Battery remaining useful life prediction	Data from literature (72 data after cleaning)	Constant current charging process model	Neural network	PKPM gives a $MARE = 3.19\%$, 44% less than ZKPM. No comparisons vs FKPM

tasks, such as process management and efficiency enhancements [41].

Looking ahead, the trajectory of PKPMs suggests a trend toward deeper integration [40,41,62,211]. Early combination methods of inputs and outputs of FKPMs and ZKPMs, are moving to an increased integration, such as in feature pre-processing, structural constraints on model architectures, and regularization techniques [72]. Greater integration necessitates increased effort and demand for specific skill sets that may not always be readily available [35]. Therefore, it becomes even more crucial to carefully weigh the trade-offs between benefits and costs [35].

Despite the potentially higher costs associated with PKPM methods, increasingly stringent regulations may serve as a catalyst for their adoption, driven by their superior accuracy and reliability [80,224]. In fact, the burgeoning need for precise predictive capabilities, particularly in regulated industries, may further incentivize the diffusion of PKPM approaches [28]. The adoption of improved models is also pulled by Industry 4.0 and 5.0 which tighten up the competitiveness toward a more customer-based production [23]. The possibility of having a more trustworthy model will allow PKPMs to be implemented in responsibility sectors, in which the use of ZKPMs is limited [104].

Crucially, the future success of PKPMs hinges on fostering closer collaboration between domain experts and data scientists [72]. As such, initiatives to bridge the gap between these two domains will be pivotal in realizing the full potential of PKPMs across diverse applications [83]. Through concerted efforts to address these challenges and capitalize on emerging opportunities, PKPMs are poised to revolutionize predictive model paradigms, paving the way for unprecedented advancements at the intersection of physics and machine learning [41].

7. Conclusions

Predictive models have been pivotal in bolstering production efficiency, product quality, scalability, and cost-effectiveness while promoting sustainability in contemporary industrial applications. To maintain a focused scope and avoid redundancy with existing literature, our review has primarily delved into the key industrial applications within a narrow context, encompassing sectors such as extraction, chemical processes, manufacturing, transportation, energy, and construction. These sectors are central to ongoing discussions on sustainability and environmental impact. Consequently, sectors like health, agriculture, utilities, and defense warrant separate comprehensive analyses and are intentionally excluded from this review. In this context, predictive models have been developed on different bases: solely based on domain-specific knowledge, exclusively on observational data, or by amalgamating both approaches, commonly referred to as Full-, Zero-, or Partial-knowledge-based predictive models, respectively.

In our review, we followed a procedure that we believe can cater to the needs of both young researchers seeking a solid foundation for their studies, industrial practitioners aiming to grasp core concepts and applications, and senior researchers seeking potential real-world applications for their findings. In fact, as a first step, we conducted a meta-review to pinpoint gaps in previously published surveys on Full-, Zero-, and Partial-knowledge-based predictive models for industrial applications. Subsequently, we presented a formal analysis of the subject matter, supplemented with illustrative examples to offer valuable insights. The core of our work comprises a review of existing research categorized by specific industrial applications. Finally, we outline the unresolved challenges and future prospects in this burgeoning field of research.

CRediT authorship contribution statement

Stefano Zampini: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Guido Parodi:** Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Validation, Visualization, Writing – original draft, Writing – review & editing. **Luca Oneto:** Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. **Andrea Coraddu:** Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. **Davide Anguita:** Conceptualization, Funding acquisition, Project administration, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Luca Oneto reports financial support was provided by European Union. Andrea Coraddu reports financial support was provided by Dutch Research Council. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is partially supported by (i) EU - NGEU National Sustainable Mobility Center (CN00000023) Italian Ministry of University and Research Decree n. 1033—17/06/2022 (Spoke 10), (ii) project SERICS (PE00000014) under the NRRP MUR program funded by the EU - NGEU, (iii) project FAIR (PE00000013) under the NRRP MUR program funded by the EU - NGEU, and (iv) project SEANERGETIC (KICH1.KICH1.21.003) of the research program Zero Emission and Circular Shipping (KIC) which is financed by the Dutch Research Council (NWO).

Table 12
Papers about PKPMs in construction industry.

Paper	Year	PKPM	Application domain	Available and/or generated data	Prior Knowledge and/or FKPM	ZKPM	Results & comparison between FKPM, ZKPM, and PKPM
[275]	2018	In (parameters tuning)	Building thermal modeling	Experimental data (measurements of 31 days)	RC model	Genetic algorithm and pattern search algorithm	The best FKPM, ZKPM, and PKPM give an average MAE of 0.7 $°C$, 0.4 $°C$, and 1.0 $°C$ respectively
[281]	2020	In	Building seismic response modeling	(i) Synthetically generated data (100 seismic sequences of 1001 data). (ii) Experimental data (23 seismic events)	Equations of motion	Neural network	(i) In the worst scenario PKPM has a correlation factor $r = 0.61$, ZKPM 0.37. (ii) The error of PKPM in predicting the displacement is $< 5\%$ with a confidence interval $> 93\%$. No comparisons vs FKPM
[282]	2021	Pre & In	Structural damage identification	(i) Synthetically generated data (100 simulations of damage cases). (ii) Experimental data (60 shaking tests)	Mechanics of structures	Neural network	(i) 97.5% of the errors of PKPM are < 0.2. 22.3% of errors of ZKPM are > 1. (ii) 93.3% of the errors of PKPM are < 0.1. 20% of errors of ZKPM are > 0.2. No comparisons vs FKPM
[276]	2022	Pre & In	Building thermal modeling	Synthetically generated data (3672 simulation samples)	Heat transfer law	Multiple linear regression	PKPM has a mean $RMSE$ of 81.312, the best ZKPM 92.836. No comparisons vs FKPM
[277]	2022	Pre & In	Building thermal modeling	Experimental data (training set of 370 days of measurements)	Laws of thermodynamics	Linear regression	In the test PKPM gives $MAE < 0.1$, while ZKPM $MAE > 0.1$. No comparisons vs FKPM
[278]	2022	Pre & In	Building thermal modeling	Experimental data (125 days, 48 samples per day)	RC model	Neural network	PKPM predicts T with and error < 0.5 $°C$, $60 - 70\%$ better than ZKPM. No comparisons vs FKPM
[279]	2023	In	Building thermal modeling	Experimental data (236 samples for each scenario)	RC model	Neural network	In the worst scenario PKPM predicts T with $MAE = 0.25$, ZKPM with 0.47. No comparisons vs FKPM
[284]	2024	Post	Tunnel settlement analysis	(i) Synthetically generated data (simulations of 10 load cases). (ii) Experimental data (12 sets of measurements available)	Multi-beam model	Neural network	(i) PKPM predicts foundation modulus with $R^2 > 0.973$ (ii) PKPM predicts settlement with $R^2 = 0.895$. No comparisons vs FKPM and ZKPM.
[283]	2024	Post	Structure full-field response modeling	(i) Synthetically generated data (time series of 12000 samples). (ii) Synthetically generated data (time series of 11360 samples)	Impulse response matrix	Convolutional neural network	(i) PKPM predicts the displacement with $RMSE = 1.41\%$, FKPM 4.37%. (ii) PKPM predicts the displacement with $RMSE = 3.61\%$. No comparisons vs ZKPM
[285]	2024	In	Structure interaction with soil	Boundary conditions	Laterally loaded pile governing equations	Neural network	PKPM is comparable to FKPM FEM simulation, but 3000 times faster. No comparisons vs ZKPM
[280]	2024	In (parameters tuning)	Building thermal modeling	Synthetically generated data (measurements of 37 days in summer and 37 in winter)	Single-zone building model	Trust Region Reflective algorithm	PKPM compared to ZKPM in different scenarios. Results are comparable. PKPM requires less data. No comparisons vs FKPM

Data availability

Data will be made available on request.

References

[1] P.N. Stearns, The Industrial Revolution in World History, Routledge, 2020.
[2] P.P. Groumpos, A critical historical and scientific overview of all industrial revolutions, IFAC-PapersOnLine 54 (13) (2021) 464–471.
[3] M.C. Zizic, M. Mladineo, N. Gjeldum, L. Celent, From industry 4.0 towards industry 5.0: A review and analysis of paradigm shift for the people, organization and technology, Energies 15 (14) (2022) 5221.
[4] L. Melnyk, O. Kubatko, I. Dehtyarova, O. Matsenko, O. Rozhko, The effect of industrial revolutions on the transformation of social and economic systems, Probl. Perspect. Manag. 17 (4) (2019) 381.
[5] T.S. Ashton, The Industrial Revolution 1760-1830, Oxford University Press, 1997.
[6] J. Mokyr, R.H. Strotz, The second industrial revolution, 1870-1914, Stor. Dell'Econ. Mond. 21945 (1) (1998).
[7] J. Greenwood, The Third Industrial Revolution: Technology, Productivity, and Income Inequality, American Enterprise Institute, 1997.
[8] K. Schwab, The Fourth Industrial Revolution, Currency, 2017.
[9] M. Xu, J.M. David, S.H. Kim, The fourth industrial revolution: Opportunities and challenges, Int. J. Financ. Res. 9 (2) (2018) 90–95.
[10] J. Muller, Enabling Technologies for Industry 5.0 - Results of a Workshop with Europe's Technology Leaders, European Commission and Directorate-General for Research and Innovation, 2020.
[11] Z. Hyder, K. Siau, F. Nah, Artificial intelligence, machine learning, and autonomous technologies in mining industry, J. Database Manag. (JDM) 30 (2) (2019) 67–79.
[12] Z.J. Baum, X. Yu, P.Y. Ayala, Y. Zhao, et al., Artificial intelligence in chemistry: current trends and future directions, J. Chem. Inf. Model. 61 (7) (2021) 3197–3212.
[13] B. Li, B. Hou, W. Yu, X. Lu, C. Yang, Applications of artificial intelligence in intelligent manufacturing: a review, Front. Inf. Technol. Electron. Eng. 18 (1) (2017) 86–96.
[14] A. Martínez-Gutiérrez, J. Díez-González, R. Ferrero-Guillén, P. Verde, R. Álvarez, H. Perez, Digital twin for automatic transportation in industry 4.0, Sensors 21 (10) (2021) 3344.
[15] O. Dudnik, M. Vasiljeva, N. Kuznetsov, M. Podzorova, et al., Trends, impacts, and prospects for implementing artificial intelligence technologies in the energy industry: the implication of open innovation, J. Open Innov. Technol. Mark. Complex. 7 (2) (2021) 155.
[16] S.O. Abioye, L.O. Oyedele, L. Akanbi, A. Ajayi, et al., Artificial intelligence in the construction industry: A review of present status, opportunities and future challenges, J. Build. Eng. 44 (2021) 103299.
[17] F. Jiang, Y. Jiang, H. Zhi, Y. Dong, et al., Artificial intelligence in healthcare: past, present and future, Stroke Vasc. Neurol. 2 (4) (2017).
[18] N.C. Eli-Chukwu, Applications of artificial intelligence in agriculture: A review, Eng. Technol. Appl. Sci. Res. 9 (4) (2019).
[19] H. Jenny, E.G. Alonso, Y. Wang, R. Minguez, Using artificial intelligence for smart water management systems, ADB Briefs (2020).
[20] R.E. Balmer, S.L. Levin, S. Schmidt, Artificial intelligence applications in telecommunications and other network industries, Telecommun. Policy 44 (6) (2020) 101977.
[21] S. De Spiegeleire, M. Maas, T. Sweijs, Artificial Intelligence and the Future of Defense: Strategic Implications for Small-And Medium-Sized Force Providers, The Hague Centre for Strategic Studies, 2017.
[22] M. Braglia, R. Gabbrielli, L. Marrazzini, L. Padellini, Key Performance Indicators and Industry 4.0–A structured approach for monitoring the implementation of digital technologies, Procedia Comput. Sci. 200 (2022) 1626–1635.
[23] X. Xu, Y. Lu, B. Vogel-Heuser, L. Wang, Industry 4.0 and Industry 5.0 - Inception, conception and perception, J. Manuf. Syst. 61 (2021) 530–535.
[24] P.K.R. Maddikunta, Q. Pham, B. Prabadevi, N. Deepa, K. Dev, T.R. Gadekallu, R. Ruby, M. Liyanage, Industry 5.0: A survey on enabling technologies and potential applications, J. Ind. Inf. Integr. 26 (2022) 100257.
[25] D.J. Horst, C.A. Duvoisin, R. de Almeida Vieira, Additive manufacturing at Industry 4.0: a review, Int. J. Eng. Tech. Res. 8 (8) (2018).
[26] S. Nambisan, D. Siegel, M. Kenney, On open innovation, platforms, and entrepreneurship, Strat. Entrep. J. 12 (3) (2018) 354–368.
[27] J. Wang, Y. Li, R.X. Gao, F. Zhang, Hybrid physics-based and data-driven models for smart manufacturing: Modelling, simulation, and explainability, J. Manuf. Syst. 63 (2022) 381–391.
[28] S. Yang, P. Navarathna, S. Ghosh, B.W. Bequette, Hybrid modeling in the era of smart manufacturing, Comput. Chem. Eng. 140 (2020) 106874.
[29] K. Lepenioti, A. Bousdekis, D. Apostolou, G. Mentzas, Prescriptive analytics: Literature review and research challenges, Int. J. Inf. Manage. 50 (2020) 57–70.
[30] R.F. Babiceanu, R. Seker, Big Data and virtualization for manufacturing cyber-physical systems: A survey of the current status and future outlook, Comput. Ind. 81 (2016) 128–137.
[31] G. Wang, A. Gunasekaran, E.W.T. Ngai, T. Papadopoulos, Big data analytics in logistics and supply chain management: Certain investigations for research and applications, Int. J. Prod. Econ. 176 (2016) 98–110.
[32] O. Merkt, On the use of predictive models for improving the quality of industrial maintenance: An analytical literature review of maintenance strategies, in: Federated Conference on Computer Science and Information Systems, 2019.
[33] P.J. Arrazola, T. Özel, D. Umbrello, M. Davies, I.S. Jawahir, Recent advances in modelling of metal machining processes, Cirp Ann. 62 (2) (2013) 695–718.
[34] K. Tsolaki, T. Vafeiadis, A. Nizamis, D. Ioannidis, D. Tzovaras, Utilizing machine learning on freight transportation and logistics applications: A review, ICT Express (2022).
[35] Y. Xu, S. Kohtz, J. Boakye, P. Gardoni, P. Wang, Physics-informed machine learning for reliability and systems safety applications: State of the art and challenges, Reliab. Eng. Syst. Saf. 230 (2023) 108900.
[36] A. Janusz, M. Grzegorowski, M. Michalak, L. Wrobel, M. Sikora, D. Slkezak, Predicting seismic events in coal mines based on underground sensor measurements, Eng. Appl. Artif. Intell. 64 (2017) 83–94.
[37] P. Backus, M. Janakiram, S. Mowzoon, C. Runger, A. Bhargava, Factory cycle-time prediction with a data-mining approach, IEEE Trans. Semicond. Manuf. 19 (2) (2006) 252–258.
[38] S. Moura, Chapter 1: Modeling and systems analysis, 2024, https://ecal.berkeley.edu/files/ce295/CH01-ModelingSystems.pdf. (Accessed 28 January 2024).
[39] R. Rai, C.K. Sahu, Driven by data or derived through physics? a review of hybrid physics guided machine learning techniques with cyber-physical system (cps) focus, IEEE Access 8 (2020) 71050–71073.
[40] J. Sansana, M.N. Joswiak, I. Castillo, Z. Wang, R. Rendall, L.H. Chiang, M.S. Reis, Recent trends on hybrid modeling for industry 4.0, Comput. Chem. Eng. 151 (2021) 107365.
[41] G.E. Karniadakis, I.G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, L. Yang, Physics-informed machine learning, Nat. Rev. Phys. 3 (6) (2021) 422–440.
[42] A. Karpatne, G. Atluri, J.H. Faghmous, M. Steinbach, A. Banerjee, A. Ganguly, S. Shekhar, N. Samatova, V. Kumar, Theory-guided data science: A new paradigm for scientific discovery from data, IEEE Trans. Knowl. Data Eng. 29 (10) (2017) 2318–2331.
[43] K. Sargsyan, H.N. Najm, R. Ghanem, On the statistical calibration of physical models, Int. J. Chem. Kinet. 47 (4) (2015) 246–276.
[44] Y. Liu, G. Glass, Effects of Mesh Density on Finite Element Analysis, Technical Report, SAE Technical Paper, 2013.
[45] M.I. Jordan, T.M. Mitchell, Machine learning: Trends, perspectives, and prospects, Science 349 (6245) (2015) 255–260.
[46] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.
[47] C.C. Aggarwal, Data Mining: the Textbook, vol. 1, 2015.
[48] R. Roscher, B. Bohn, M.F. Duarte, J. Garcke, Explainable machine learning for scientific insights and discoveries, IEEE Access 8 (2020) 42200–42216.
[49] A. Cozad, N.V. Sahinidis, D.C. Miller, Learning surrogate models for simulation-based optimization, AIChE J. 60 (6) (2014) 2211–2227.
[50] R.T.Q. Chen, Y. Rubanova, J. Bettencourt, D.K. Duvenaud, Neural ordinary differential equations, in: Neural Information Processing Systems, 2018.
[51] R. Liu, S. Liu, X. Zhang, A physics-informed machine learning model for porosity analysis in laser powder bed fusion additive manufacturing, Int. J. Adv. Manuf. Technol. 113 (7–8) (2021) 1943–1958.
[52] D. Joenssen, A.S. Hada, J. Lenz, Physics-informed imputation for data cleaning and pre-processing in robust smart manufacturing systems, Procedia Comput. Sci. 232 (2024) 377–387.
[53] M. Tembely, A.M. AlSumaiti, W.S. Alameri, Machine and deep learning for estimating the permeability of complex carbonate rock from X-ray micro-computed tomography, Energy Rep. 7 (2021) 1460–1472.
[54] G. Parodi, L. Oneto, G. Ferro, S. Zampini, M. Robba, D. Anguita, A. Coraddu, Physics informed data driven techniques for power flow analysis, in: IEEE Symposium Series on Computational Intelligence, 2023.
[55] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, et al., Multipole graph neural operator for parametric partial differential equations, Adv. Neural Inf. Process. Syst. 33 (2020) 6755–6766.
[56] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, et al., Neural operator: Graph kernel network for partial differential equations, 2020, arXiv preprint arXiv:2003.03485.
[57] L. Lu, P. Jin, G. Pang, Z. Zhang, G.E. Karniadakis, Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators, Nat. Mach. Intell. 3 (3) (2021) 218–229.
[58] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations, 2017, arXiv preprint arXiv:1711.10561.
[59] J. Kim, C. Luettgen, K. Paynabar, F. Boukouvala, Physics-based penalization for hyperparameter estimation in gaussian process regression, Comput. Chem. Eng. 178 (2023) 108320.

[60] J.X. Wang, J.L. Wu, H. Xiao, Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on DNS data, Phys. Rev. Fluids 2 (3) (2017) 034603.
[61] M. Brosowsky, F. Keck, O. Dünkel, M. Zöllner, Sample-specific output constraints for neural networks, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2021.
[62] A. Foucquier, S. Robert, F. Suard, L. Stéphan, A. Jay, State of the art in building modelling and energy performances prediction: A review, Renew. Sustain. Energy Rev. 23 (2013) 272–288.
[63] M. Von Stosch, R. Oliveira, J. Peres, S.F. de Azevedo, Hybrid semi-parametric modeling in process systems engineering: Past, present and future, Comput. Chem. Eng. 60 (2014) 86–101.
[64] X. Li, J. Wen, Review of building energy modeling for control and operation, Renew. Sustain. Energy Rev. 37 (2014) 517–537.
[65] M. Zhou, H. Jin, W. Wang, A review of vehicle fuel consumption models to evaluate eco-driving and eco-routing, Transp. Res. Part D: Transp. Environ. 49 (2016) 203–218.
[66] S. Zendehboudi, N. Rezaei, A. Lohi, Applications of hybrid models in chemical, petroleum, and energy systems: A systematic review, Appl. Energy 228 (2018) 2539–2566.
[67] A. Boodi, K. Beddiar, M. Benamour, Y. Amirat, M. Benbouzid, Intelligent systems for building energy and occupant comfort optimization: A state of the art review and recommendations, Energies 11 (10) (2018) 2604.
[68] L. von Rueden, S. Mayer, R. Sifa, C. Bauckhage, J. Garcke, Combining machine learning and simulation to a hybrid modelling approach: Current and future directions, in: International Symposium on Intelligent Data Analysis, 2020.
[69] A.A.A. Gassar, S.H. Cha, Energy prediction techniques for large-scale buildings towards a sustainable built environment: A review, Energy Build. 224 (2020) 110238.
[70] J. Willard, X. Jia, S. Xu, M. Steinbach, V. Kumar, Integrating physics-based modeling with machine learning: A survey, 2020, arXiv preprint arXiv:2003.04919.
[71] Y. Li, Z. O'Neill, L. Zhang, J. Chen, P. Im, J. DeGraw, Grey-box modeling and application for building energy simulations-A critical review, Renew. Sustain. Energy Rev. 146 (2021) 111174.
[72] L. von Rueden, S. Mayer, K. Beckh, B. Georgiev, S. Giesselbach, R. Heese, B. Kirsch, J. Pfrommer, A. Pick, R. Ramamurthy, M. Walczak, J. Garcke, C. Bauckhage, S. J., Informed machine learning-a taxonomy and survey of integrating prior knowledge into learning systems, IEEE Trans. Knowl. Data Eng. 35 (1) (2021) 614–633.
[73] S. Tamilselvi, S. Gunasundari, N. Karuppiah, A. Razak RK, S. Madhusudan, V.M. Nagarajan, T. Sathish, M.Z.M. Shamim, C.A. Saleel, A. Afzal, A review on battery modelling techniques, Sustainability 13 (18) (2021) 10042.
[74] M. Mozaffar, S. Liao, X. Xie, S. Saha, C. Park, J. Cao, W.K. Liu, Z. Gan, Mechanistic artificial intelligence (mechanistic-AI) for modeling, design, and control of advanced manufacturing processes: Current state and perspectives, J. Mater. Process. Technol. 302 (2022) 117485.
[75] S.R. Vadyala, S.N. Betgeri, J.C. Matthews, E. Matthews, A review of physics-based machine learning in civil engineering, Results Eng. 13 (2022) 100316.
[76] N. Sharma, Y.A. Liu, A hybrid science-guided machine learning approach for modeling chemical processes: A review, AIChE J. 68 (5) (2022) e17609.
[77] B. Huang, J. Wang, Applications of physics-informed neural networks in power systems-a review, IEEE Trans. Power Syst. 38 (1) (2022) 572–588.
[78] W. Bradley, J. Kim, Z. Kilwein, L. Blakely, M. Eydenberg, J. Jalvin, C. Laird, F. Boukouvala, Perspectives on the integration between first-principles and data-driven modeling, Comput. Chem. Eng. 166 (2022) 107898.
[79] S. Cuomo, V.S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, F. Piccialli, Scientific machine learning through physics-informed neural networks: Where we are and what's next, J. Sci. Comput. 92 (3) (2022) 88.
[80] A. Fan, J. Yang, L. Yang, D. Wu, N. Vladimir, A review of ship fuel consumption models, Ocean Eng. 264 (2022) 112405.
[81] Z. Hao, S. Liu, Y. Zhang, C. Ying, Y. Feng, H. Su, J. Zhu, Physics-informed machine learning: A survey on problems, methods and applications, 2022, arXiv preprint arXiv:2211.08064.
[82] W. Guo, Z. Sun, S.B. Vilsen, J. Meng, D.I. Stroe, Review of "grey box" lifetime modeling for lithium-ion battery: Combining physics and data-driven methods, J. Energy Storage 56 (2022) 105992.
[83] J. Willard, X. Jia, S. Xu, M. Steinbach, V. Kumar, Integrating scientific knowledge with machine learning for engineering and environmental systems, ACM Comput. Surv. 55 (4) (2022) 1–37.
[84] G. Zhong, X. Ling, L. Wang, From shallow feature learning to deep learning: Benefits from the width and depth of deep architectures, Wiley Interdiscip. Rev. Data Min. Knowl. Discov. 9 (1) (2019) e1255.
[85] N.L. Hjort, G. Claeskens, Frequentist model average estimators, J. Amer. Statist. Assoc. 98 (464) (2003) 879–899.
[86] D.V. Lindley, Bayesian Statistics: A Review, SIAM, 1972.
[87] M. Aly, Survey on multiclass classification methods, Neural Netw. 19 (1–9) (2005) 2.
[88] M. Fernández-Delgado, M.S. Sirsat, E. Cernadas, S. Alawadi, et al., An extensive experimental survey of regression methods, Neural Netw. 111 (2019) 11–34.
[89] W.H. Joerding, J.L. Meador, Encoding a priori information in feedforward networks, Neural Netw. 4 (6) (1991) 847–856.
[90] D.C. Psichogios, L.H. Ungar, A hybrid neural network-first principles approach to process modeling, AIChE J. 38 (10) (1992) 1499–1511.
[91] M. Agarwal, Combining neural and conventional paradigms for modeling, prediction, and control, in: International Conference on Control Applications, 1995.
[92] L. Oneto, D. Chicco, Eight quick tips for biologically and medically informed machine learning, PLoS Comput. Biol. 21 (1) (2025) e1012711.
[93] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics informed deep learning (part ii): Data-driven discovery of nonlinear partial differential equations. arxiv 2017, 2017, arXiv preprint arXiv:1711.10566.
[94] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 378 (2019) 686–707.
[95] S. Shalev-Shwartz, S. Ben-David, Understanding Machine Learning: From Theory to Algorithms, Cambridge University Press, 2014.
[96] D. Bacciu, F. Errica, A. Micheli, M. Podda, A gentle introduction to deep learning for graphs, Neural Netw. 129 (2020) 203–221.
[97] D. Bacciu, A. Micheli, A. Sperduti, Compositional generative mapping for tree-structured data–part I: Bottom-up probabilistic modeling of trees, IEEE Trans. Neural Netw. Learn. Syst. 23 (12) (2012) 1987–2002.
[98] C.C. Aggarwal, Neural Networks and Deep Learning: a Textbook, Spinger, 2023.
[99] OpenAI, GPT-4 technical report, 2023, arXiv preprint arXiv:2303.08774.
[100] B. Schölkopf, Causality for machine learning, in: Probabilistic and Causal Inference: The Works of Judea Pearl, 2022.
[101] T. Vigen, Spurious Correlations, Hachette UK, 2015.
[102] X. Zhu, X. Wu, Cost-constrained data acquisition for intelligent data preparation, IEEE Trans. Knowl. Data Eng. 17 (11) (2005) 1542–1556.
[103] G. Benton, M. Finzi, P. Izmailov, A.G. Wilson, Learning invariances in neural networks from training data, in: Neural Information Processing Systems, 2020.
[104] E. Giunchiglia, F. Imrie, M. Van Der Schaar, T. Lukasiewicz, Machine learning with requirements: a manifesto, 2023, arXiv preprint arXiv:2304.03674.
[105] S. Sagiroglu, D. Sinanc, Big data: A review, in: 2013 International Conference on Collaboration Technologies and Systems, CTS, 2013.
[106] J.E. Van Engelen, H.H. Hoos, A survey on semi-supervised learning, Mach. Learn. 109 (2) (2020) 373–440.
[107] A. Coraddu, L. Oneto, D. Ilardi, S. Stoumpos, G. Theotokatos, Marine dual fuel engines monitoring in the wild through weakly supervised data analytics, Eng. Appl. Artif. Intell. 100 (2021) 104179.
[108] F. Cipollini, L. Oneto, A. Coraddu, A.J. Murphy, D. Anguita, Condition-based maintenance of naval propulsion systems: Data analysis with minimal feedback, Reliab. Eng. Syst. Saf. 177 (2018) 12–23.
[109] L. Guan, Preparation of future weather data to study the impact of climate change on buildings, Build. Environ. 44 (4) (2009) 793–800.
[110] M. Banbura, D. Giannone, L. Reichlin, Nowcasting, 2010, ECB Working Paper.
[111] B. Abraham, J. Ledolter, Statistical methods for forecasting, John Wiley & Sons, 2009.
[112] K.H. Dreborg, Essence of backcasting, Futures 28 (9) (1996) 813–828.
[113] R.E. Nieman, D.G. Fisher, D.E. Seborg, A review of process identification and parameter estimation techniques, Internat. J. Control 13 (2) (1971) 209–264.
[114] A.B. Jiménez, J.L. Lázaro, J.R. Dorronsoro, Innovations in Hybrid Intelligent Systems, Springer Berlin Heidelberg, 2007.
[115] M.Z. Naser, A.H. Alavi, Error metrics and performance fitness indicators for artificial intelligence and machine learning in engineering and sciences, Archit. Struct. Constr. (2021) 1–19.
[116] L. Miglianti, F. Cipollini, L. Oneto, G. Tani, et al., Predicting the cavitating marine propeller noise at design stage: A deep learning based approach, Ocean Eng. 209 (2020) 107481.
[117] P.P. Angelov, E.A. Soares, R. Jiang, N.I. Arnold, P.M. Atkinson, Explainable artificial intelligence: an analytical review, Wiley Interdiscip. Rev. Data Min. Knowl. Discov. 11 (5) (2021) e1424.
[118] H. White, A reality check for data snooping, Econometrica 68 (5) (2000) 1097–1126.
[119] M. Fernández-Delgado, E. Cernadas, S. Barro, D. Amorim, Do we need hundreds of classifiers to solve real world classification problems? J. Mach. Learn. Res. 15 (1) (2014) 3133–3181.
[120] M. Wainberg, B. Alipanahi, B.J. Frey, Are random forests truly the best classifiers? J. Mach. Learn. Res. 17 (1) (2016) 3837–3841.
[121] L. Oneto, Model Selection and Error Estimation in a Nutshell, Springer, 2020.
[122] L. Cao, Beyond iid: Non-iid thinking, informatics, and learning, IEEE Intell. Syst. 37 (4) (2022) 5–17.
[123] N.R. Draper, F. Pukelsheim, An overview of design of experiments, Statist. Papers 37 (1996) 1–32.
[124] J. Hamilton, Time Series Analysis, Princeton University Press, 2020.
[125] G. Chandrashekar, F. Sahin, A survey on feature selection methods, Comput. Electr. Eng. 40 (1) (2014) 16–28.

[126] D.H. Wolpert, The supervised learning no-free-lunch theorems, in: Soft Computing and Industry, 2002.
[127] J. Shawe-Taylor, N. Cristianini, Kernel Methods for Pattern Analysis, Cambridge University Press, 2004.
[128] Z.H. Zhou, Ensemble Methods: Foundations and Algorithms, CRC Press, 2012.
[129] D. Barber, Bayesian Reasoning and Machine Learning, Cambridge University Press, 2012.
[130] J. Lu, G. Ma, G. Zhang, Fuzzy machine learning: A comprehensive framework and systematic review, IEEE Trans. Fuzzy Syst. 32 (7) (2024) 3861–3878.
[131] M. Woźniak, J. Szczotka, A. Sikora, A. Zielonka, Fuzzy logic type-2 intelligent moisture control system, Expert Syst. Appl. 238 (2024) 121581.
[132] K. Weiss, T.M. Khoshgoftaar, D.D. Wang, A survey of transfer learning, J. Big Data 3 (1) (2016) 1–40.
[133] Q. Yang, Y. Zhang, W. Dai, S.J. Pan, Transfer Learning, Cambridge University Press, 2020.
[134] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016.
[135] O. Bousquet, A. Elisseeff, Stability and generalization, J. Mach. Learn. Res. 2 (2002) 499–526.
[136] L. Oneto, A. Ghio, S. Ridella, D. Anguita, Fully empirical and data-dependent stability-based bounds, IEEE Trans. Cybern. 45 (9) (2014) 1913–1926.
[137] P. Klesk, M. Korzen, Sets of approximating functions with finite Vapnik–Chervonenkis dimension for nearest-neighbors algorithms, Pattern Recognit. Lett. 32 (14) (2011) 1882–1893.
[138] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.
[139] M.E. Celebi, K. Aydin, Unsupervised Learning Algorithms, Springer, 2016.
[140] Z. Ghahramani, Unsupervised Learning, Springer, 2003.
[141] Y. Bengio, A.C. Courville, P. Vincent, Unsupervised feature learning and deep learning: A review and new perspectives, 2012, p. 2012, 1 (2665) CoRR, abs/1206.5538.
[142] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, D. Pedreschi, A survey of methods for explaining black box models, ACM Comput. Surv. 51 (5) (2018) 1–42.
[143] A.B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, et al., Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI, Inf. Fusion 58 (2020) 82–115.
[144] P. Naga Srinivasu, M.F. Ijaz, M. Woźniak, XAI-driven model for crop recommender system for use in precision agriculture, Comput. Intell. 40 (1) (2024) e12629.
[145] J. Gou, B. Yu, S.J. Maybank, D. Tao, Knowledge distillation: A survey, Int. J. Comput. Vis. 129 (6) (2021) 1789–1819.
[146] K. Wang, N. Fang, Z. Huang, J. Xu, S. Zhang, J. Qin, A physics-informed knowledge distillation model with spatial–temporal attention for energy consumption pre-assessment in sustainable additive manufacturing, Sustain. Energy Technol. Assess. 60 (2023) 103537.
[147] S. Liu, B.B. Kappes, B. Amin-ahmadi, O. Benafan, X. Zhang, A.P. Stebner, Physics-informed machine learning for composition-process-property design: Shape memory alloy demonstration, Appl. Mater. Today 22 (2021) 100898.
[148] J.P.C. Kleijnen, Design of experiments: overview, in: 2008 Winter Simulation Conference, 2008.
[149] N. Gupta, H. Patel, S. Afzal, N. Panwar, et al., Data quality toolkit: Automatic assessment of data quality and remediation for machine learning datasets, 2021, arXiv preprint arXiv:2108.05935.
[150] M.T. Ribeiro, S. Singh, C. Guestrin, "Why should I trust you?" Explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016.
[151] A. Altmann, L. Toloşi, O. Sander, T. Lengauer, Permutation importance: a corrected feature importance measure, Bioinformatics 26 (10) (2010) 1340–1347.
[152] L.W. Cai, Fundamentals of Mechanical Vibrations, John Wiley & Sons, 2016.
[153] B.P. Flannery, W.H. Press, S.A. Teukolsky, W. Vetterling, Numerical Recipes in C, vol. 24, (no. 78) Press Syndicate of the University of Cambridge, New York, 1992, p. 36.
[154] L. Oneto, S. Ridella, D. Anguita, Do we really need a new theory to understand over-parameterization? Neurocomputing 543 (2023) 126227.
[155] C. Qi, Big data management in the mining industry, Int. J. Miner. Met. Mater. 27 (2) (2020) 131–139.
[156] F.P. Carvalho, Mining industry and sustainable development: time for change, Food Energy Secur. 6 (2) (2017) 61–77.
[157] L.A.M. Castro, R.P. Bewick, T.G. Carter, An overview of numerical modelling applied to deep mining, Innov. Numer. Model. Geomech. (2012) 393–414.
[158] S. Rajendran, T.R. Vakamalla, N. Mangadoddy, Numerical methods in mineral processing: an overview, Miner. Process. (2023) 251–285.
[159] G. Beer, C. Dunser, M. Noronha, Recent advances in the numerical simulation of tunnel excavation, in: ISRM Congress, 2003.
[160] K. Balaji, M. Rabiei, V. Suicmez, C.H. Canbaz, et al., Status of data-driven methods and their applications in oil and gas industry, in: SPE Europec Featured At EAGE Conference and Exhibition? 2018.
[161] M. Jooshaki, A. Nad, S. Michaux, A systematic review on the application of machine learning in exploiting mineralogical data in mining and mineral industry, Minerals 11 (8) (2021) 816.
[162] H. Zhao, Z. Kang, X. Zhang, H. Sun, et al., A physics-based data-driven numerical model for reservoir history matching and prediction with a field application, SPE J. 21 (06) (2016) 2175–2194.
[163] H. Zhao, Y. Li, S. Cui, G. Shang, A.C. Reynolds, Z. Guo, H.A. Li, History matching and production optimization of water flooding based on a data-driven interwell numerical simulation model, J. Nat. Gas Sci. Eng. 31 (2016) 48–66.
[164] Z. Guo, A.C. Reynolds, H. Zhao, A physics-based data-driven model for history-matching, prediction and characterization of waterflooding performance, in: SPE Reservoir Simulation Conference? 2017.
[165] Z. Guo, A.C. Reynolds, H. Zhao, Waterflooding optimization with the INSIM-FT data-driven model, Comput. Geosci. 22 (2018) 745–761.
[166] Y. Zhang, J. He, C. Yang, J. Xie, et al., A physics-based data-driven model for history matching, prediction, and characterization of unconventional reservoirs, SPE J. 23 (04) (2018) 1105–1125.
[167] Y. Li, M. Onur, INSIM-BHP: A physics-based data-driven reservoir model for history matching and forecasting with bottomhole pressure and production rate data under waterflooding, J. Comput. Phys. 473 (2023) 111714.
[168] Z.L. Jin, Y. Liu, L.J. Durlofsky, Deep-learning-based surrogate model for reservoir simulation with time-varying well controls, J. Pet. Sci. Eng. 192 (2020) 107273.
[169] D.R. Harp, D. O'Malley, B. Yan, R. Pawar, On the feasibility of using physics-informed machine learning for underground reservoir pressure management, Expert Syst. Appl. 178 (2021) 115006.
[170] M.M. Almajid, M.O. Abu-Al-Saud, Prediction of porous media fluid flow using physics informed neural networks, J. Pet. Sci. Eng. 208 (2022) 109205.
[171] K. Zhang, Y. Zuo, H. Zhao, X. Ma, et al., Fourier neural operator for solving subsurface oil/water two-phase flow partial differential equation, Spe J. 27 (03) (2022) 1815–1830.
[172] M.S.F. Bangi, J.S. Kwon, Deep hybrid modeling of chemical process: Application to hydraulic fracturing, Comput. Chem. Eng. 134 (2020) 106696.
[173] C.A. Heaton, The Chemical Industry, Springer Science & Business Media, 1993.
[174] E. Johnson, Sustainability in the Chemical Industry, Springer Science & Business Media, 2012.
[175] R.B. Bird, Transport Phenomena, Wiley, 1960.
[176] B.W. Bequette, Process Dynamics: Modeling Analysis and Simulation, Prentice Hall Incorporated, 1998.
[177] I.T. Cameron, K. Hangos, Process Modelling and Model Analysis, Elsevier, 2001.
[178] J.A. Moulijn, M. Makkee, A.E. Van Diepen, Chemical Process Technology, John Wiley & Sons, 2013.
[179] Z. Ge, Z. Song, F. Gao, Review of recent research on data-based process monitoring, Ind. Eng. Chem. Res. 52 (10) (2013) 3543–3562.
[180] M.S. Reis, G. Gins, Industrial process monitoring in the big data/industry 4.0 era: From detection to diagnosis, to prognosis, Processes 5 (3) (2017) 35.
[181] M.S. Reis, P.M. Saraiva, Data-centric process systems engineering for the chemical industry 4.0, in: Systems Engineering in the Fourth Industrial Revolution, 2019, pp. 137–159.
[182] B. Sanchez-Lengeling, A. Aspuru-Guzik, Inverse molecular design using machine learning: Generative models for matter engineering, Science 361 (6400) (2018) 360–365.
[183] R. Gómez-Bombarelli, J.N. Wei, D. Duvenaud, J.M. Hernández-Lobato, et al., Automatic chemical design using a data-driven continuous representation of molecules, ACS Central Sci. 4 (2) (2018) 268–276.
[184] K.T. Butler, D.W. Davies, H. Cartwright, O. Isayev, A. Walsh, Machine learning for molecular and materials science, Nature 559 (7715) (2018) 547–555.
[185] S.I. Ngo, Y. Lim, Solution and parameter identification of a fixed-bed reactor model for catalytic CO2 methanation using physics-informed neural networks, Catalysts 11 (11) (2021) 1304.
[186] L. Scheffold, T. Finkler, U. Piechottka, Gray-box system modeling using symbolic regression and nonlinear model predictive control of a semibatch polymerization, Comput. Chem. Eng. 146 (2021) 107204.
[187] F. Vega-Ramon, X. Zhu, T.R. Savage, P. Petsagkourakis, K. Jing, D. Zhang, Kinetic and hybrid modeling for yeast astaxanthin production under uncertainty, Biotechnol. Bioeng. 118 (12) (2021) 4854–4866.
[188] K. Li, H. Duan, L. Liu, R. Qiu, B. van den Akker, et al., An integrated first principal and deep learning approach for modeling nitrous oxide emissions from wastewater treatment plants, Environ. Sci. Technol. 56 (4) (2022) 2816–2826.
[189] T. Asrav, E. Aydin, Physics-informed recurrent neural networks and hyper-parameter optimization for dynamic process systems, Comput. Chem. Eng. 173 (2023) 108195.
[190] D.R. Cassar, ViscNet: Neural network for predicting the fragility index and the temperature-dependency of viscosity, Acta Mater. 206 (2021) 116602.
[191] I. Chiniforooshan Esfahani, A data-driven physics-informed neural network for predicting the viscosity of nanofluids, AIP Adv. 13 (2) (2023).
[192] H. Narayanan, T. Seidler, M.F. Luna, M. Sokolov, M. Morbidelli, A. Butté, Hybrid models for the simulation and prediction of chromatographic processes for protein capture, J. Chromatogr. A 1650 (2021) 462248.

[193] J. Ren, D. Liu, Y. Wan, Modeling and application of czochralski silicon single crystal growth process using hybrid model of data-driven and mechanism-based methodologies, J. Process Control 104 (2021) 74–85.
[194] M.S.F. Bangi, K. Kao, J.S. Kwon, Physics-informed neural networks for hybrid modeling of lab-scale batch fermentation for β-carotene production using saccharomyces cerevisiae, Chem. Eng. Res. Des. 179 (2022) 415–423.
[195] P. Shah, M.Z. Sheriff, M.S.F. Bangi, C. Kravaris, J.S. Kwon, C. Botre, J. Hirota, Deep neural network-based hybrid modeling and experimental validation for an industry-scale fermentation process: Identification of time-varying dependencies among parameters, Chem. Eng. J. 441 (2022) 135643.
[196] A. Gaikwad, R. Yavari, M. Montazeri, K. Cole, L. Bian, P. Rao, Toward the digital twin of additive manufacturing: Integrating thermal simulations, sensing, and analytics to detect process faults, Iise Trans. 52 (11) (2020) 1204–1217.
[197] N. Zobeiry, J. Reiner, R. Vaziri, Theory-guided machine learning for damage characterization of composites, Compos. Struct. 246 (2020) 112407.
[198] K. Shukla, P.C. Di Leoni, J. Blackshire, D. Sparkman, G.E. Karniadakis, Physics-informed neural network for ultrasound nondestructive quantification of surface breaking cracks, J. Nondestruct. Eval. 39 (2020) 1–20.
[199] Q. Zhu, Z. Liu, J. Yan, Machine learning for metal additive manufacturing: predicting temperature and melt pool fluid dynamics using physics-informed neural networks, Comput. Mech. 67 (2021) 619–635.
[200] N. Zobeiry, K.D. Humfeld, A physics-informed machine learning approach for solving heat transfer equation in advanced manufacturing and engineering applications, Eng. Appl. Artif. Intell. 101 (2021) 104232.
[201] C. Rao, H. Sun, Y. Liu, Physics-informed deep learning for computational elastodynamics without labeled data, J. Eng. Mech. 147 (8) (2021) 04021043.
[202] S.A. Niaki, E. Haghighat, T. Campbell, A. Poursartip, R. Vaziri, Physics-informed neural network for modelling the thermochemical curing process of composite-tool systems during manufacture, Comput. Methods Appl. Mech. Engrg. 384 (2021) 113959.
[203] E. Zhang, M. Dao, G.E. Karniadakis, S. Suresh, Analyses of internal structures and defects in materials using physics-informed neural networks, Sci. Adv. 8 (7) (2022) eabk0644.
[204] Y. Li, J. Wang, Z. Huang, R.X. Gao, Physics-informed meta learning for machining tool wear prediction, J. Manuf. Syst. 62 (2022) 17–27.
[205] E. Salvati, A. Tognan, L. Laurenti, M. Pelegatti, F. De Bona, A defect-based physics-informed machine learning framework for fatigue finite life prediction in additive manufacturing, Mater. Des. 222 (2022) 111089.
[206] T. Zhou, S. Jiang, T. Han, S. Zhu, Y. Cai, A physically consistent framework for fatigue life prediction using probabilistic physics-informed neural network, Int. J. Fatigue 166 (2023) 107234.
[207] S. Niu, E. Zhang, Y. Bazilevs, V. Srivastava, Modeling finite-strain plasticity using physics-informed neural network and assessment of the network performance, J. Mech. Phys. Solids 172 (2023) 105177.
[208] J. He, C. Chadha, S. Kushwaha, S. Koric, D. Abueidda, I. Jasiuk, Deep energy method in topology optimization applications, Acta Mech. 234 (4) (2023) 1365–1379.
[209] H. Jeong, J. Bai, C.P. Batuwatta-Gamage, C. Rathnayaka, et al., A physics-informed neural network-based topology optimization (PINNTO) framework for structural optimization, Eng. Struct. 278 (2023) 115484.
[210] A. Ciampaglia, A. Tridello, D.S. Paolino, F. Berto, Data driven method for predicting the effect of process parameters on the fatigue response of additive manufactured AlSi10Mg parts, Int. J. Fatigue 170 (2023) 107500.
[211] L. Wang, S. Zhu, C. Luo, D. Liao, Q. Wang, Physics-guided machine learning frameworks for fatigue life prediction of AM materials, Int. J. Fatigue 172 (2023) 107658.
[212] A. Coraddu, L. Oneto, F. Baldi, D. Anguita, Ship efficiency forecast based on sensors data collection: Improving numerical models through data analytics, in: OCEANS 2015, Genova, 2015.
[213] X. Wang, Z. Zou, L. Yu, W. Cai, System identification modeling of ship manoeuvring motion in 4 degrees of freedom based on support vector machines, China Ocean Eng. 29 (4) (2015) 519–534.
[214] X. Wang, Z. Zou, X. Hou, F. Xu, System identification modelling of ship manoeuvring motion based on ϵ-support vector regression, J. Hydrodyn. 27 (4) (2015) 502–512.
[215] A. Coraddu, L. Oneto, F. Baldi, D. Anguita, Vessels fuel consumption forecast and trim optimisation: A data analytics perspective, Ocean Eng. 130 (2017) 351–370.
[216] L. Yang, G. Chen, N.G.M. Rytter, J. Zhao, D. Yang, A genetic algorithm-based grey-box model for ship fuel consumption prediction towards sustainable shipping, Ann. Oper. Res. (2019) 1–27.
[217] A. Dourado, F.A.C. Viana, Physics-informed neural networks for missing physics estimation in cumulative damage models: a case study in corrosion fatigue, J. Comput. Inf. Sci. Eng. 20 (6) (2020) 061007.
[218] A.J. Huang, S. Agarwal, Physics informed deep learning for traffic state estimation, in: 2020 IEEE 23rd International Conference on Intelligent Transportation Systems, ITSC, 2020.
[219] R. Yan, S. Wang, J. Cao, D. Sun, Shipping domain knowledge informed prediction and optimization in port state control, Transp. Res. Part B: Methodol. 149 (2021) 52–78.
[220] T. Wang, G. Li, L.I. Hatledal, R. Skulstad, V. Æ søy, H. Zhang, Incorporating approximate dynamics into data-driven calibrator: A representative model for ship maneuvering prediction, IEEE Trans. Ind. Inform. 18 (3) (2021) 1781–1789.
[221] R.G. Nascimento, M. Corbetta, C.S. Kulkarni, F.A.C. Viana, Hybrid physics-informed neural networks for lithium-ion battery modeling and prognosis, J. Power Sources 513 (2021) 230526.
[222] H. Pang, L. Wu, J. Liu, X. Liu, K. Liu, Physics-informed neural network approach for heat generation rate estimation of lithium-ion battery under various driving conditions, J. Energy Chem. 78 (2023) 1–12.
[223] W.Q. Hao, L. Tan, X.G. Yang, D.Q. Shi, M.L. Wang, G.L. Miao, Y.S. Fan, A physics-informed machine learning approach for notch fatigue evaluation of alloys used in aerospace, Int. J. Fatigue 170 (2023) 107536.
[224] T. Vairo, D. Cademartori, D. Clematis, M.P. Carpanese, B. Fabiano, Solid oxide fuel cells for shipping: A machine learning model for early detection of hazardous system deviations, Process. Saf. Environ. Prot. 172 (2023) 184–194.
[225] X. Lang, D. Wu, W. Mao, Physics-informed machine learning models for ship speed prediction, Expert Syst. Appl. 238 (2024) 121877.
[226] J. Lee, E. Lapira, B. Bagheri, H. Kao, Recent advances and trends in predictive manufacturing systems in big data environment, Manuf. Lett. 1 (1) (2013) 38–41.
[227] A. Khan, K. Turowski, A survey of current challenges in manufacturing industry and preparation for industry 4.0, in: Proceedings of the First International Scientific Conference "Intelligent Information Technologies for Industry", Vol. 1, IITI'16, 2016.
[228] N. Roussel, J. Spangenberg, J. Wallevik, R. Wolfs, Numerical simulations of concrete processing: From standard formative casting to additive manufacturing, Cem. Concr. Res. 135 (2020) 106075.
[229] M.C. Kuşhan, Ö. Poyraz, Y. Uzononat, O. Sezan, Systematical review on the numerical simulations of laser powder bed additive manufacturing, Sigma J. Eng. Nat. Sci. 36 (4) (2018) 1197–1214.
[230] T. Wuest, D. Weimer, C. Irgens, K. Thoben, Machine learning in manufacturing: advantages, challenges, and applications, Prod. Manuf. Res. 4 (1) (2016) 23–45.
[231] R. Rahul, K.T. Manoj, I. Dmitry, D. Alexandre, Machine learning in manufacturing and industry 4.0 applications, Int. J. Prod. Res. 59 (16) (2021) 4773–4778.
[232] A. Dogan, D. Birant, Machine learning and data mining in manufacturing, Expert Syst. Appl. 166 (2021) 114060.
[233] J. Rodrigue, Transportation and globalization, Encycl. of (2007).
[234] T. Wiesenthal, A. Condeço-Melhorado, G. Leduc, Innovation in the European transport sector: A review, Transp. Policy 42 (2015) 86–93.
[235] M. Ramirez-Peña, P.F. Mayuet, J.M. Vazquez-Martinez, M. Batista, Sustainability in the aerospace, naval, and automotive supply chain 4.0: Descriptive review, Materials 13 (24) (2020) 5625.
[236] V.A. Wankhede, S. Vinodh, State of the art review on industry 4.0 in manufacturing with the focus on automotive sector, Int. J. Lean Six Sigma 13 (3) (2022) 692–732.
[237] A. Zielonka, A. Sikora, M. Woźniak, Fuzzy rules intelligent car real-time diagnostic system, Eng. Appl. Artif. Intell. 135 (2024) 108648.
[238] M. Woźniak, A. Zielonka, A. Sikora, Driving support by type-2 fuzzy logic control model, Expert Syst. Appl. 207 (2022) 117798.
[239] I. de la Peña Zarzuelo, M.J.F. Soeane, B.L. Bermúdez, Industry 4.0 in the port and maritime industry: A literature review, J. Ind. Inf. Integr. 20 (2020) 100173.
[240] I. Guyon, R. Amine, S. Tamayo, F. Fontane, Analysis of the opportunities of industry 4.0 in the aeronautical sector, in: 10th International Multi-Conference on Complexity, Informatics and Cybernetics, IMCIC 2019, 2019.
[241] C. Laiton-Bonadiez, J.W. Branch-Bedoya, J. Zapata-Cortes, E. Paipa-Sanabria, M. Arango-Serna, Industry 4.0 technologies applied to the rail transportation industry: a systematic review, Sensors 22 (7) (2022) 2491.
[242] S.P. Shepherd, A review of system dynamics models applied in transportation, Transp. B: Transp. Dyn. 2 (2) (2014) 83–105.
[243] H. Nguyen, L. Kieu, T. Wen, C. Cai, Deep learning methods in transportation domain: a review, IET Intell. Transp. Syst. 12 (9) (2018) 998–1004.
[244] G. Cademartori, L. Oneto, F. Valdenazzi, A. Coraddu, A. Gambino, D. Anguita, A review on ship motions and quiescent periods prediction models, Ocean Eng. 280 (2023) 114822.
[245] K. Zhou, C. Fu, S. Yang, Big data driven smart energy management: From big data to big insights, Renew. Sustain. Energy Rev. 56 (2016) 215–225.
[246] T. Kurbatova, T. Perederii, Global trends in renewable energy development, in: 2020 IEEE KhPI Week on Advanced Technology, KhPIWeek, 2020.
[247] A. Kalair, N. Abas, M.S. Saleem, A.R. Kalair, N. Khan, Role of energy storage systems in energy transition from fossil fuels to renewables, Energy Storage 3 (1) (2021) e135.
[248] S. Kakac, A. Pramuanjaroenkij, X.Y. Zhou, A review of numerical modeling of solid oxide fuel cells, Int. J. Hydrog. Energy 32 (7) (2007) 761–786.
[249] J. Antonanzas, N. Osorio, R. Escobar, R. Urraca, et al., Review of photovoltaic power forecasting, Sol. Energy 136 (2016) 78–111.
[250] F. Alobaid, N. Mertens, R. Starkloff, T. Lanz, et al., Progress in dynamic simulation of thermal power plants, Prog. Energy Combust. Sci. 59 (2017) 79–162.

[251] H. Wang, Z. Lei, X. Zhang, B. Zhou, J. Peng, A review of deep learning for renewable energy forecasting, Energy Convers. Manage. 198 (2019) 111799.
[252] Y. Li, K. Liu, A.M. Foley, A. Zülke, M. Berecibar, et al., Data-driven health estimation and lifetime prediction of lithium-ion batteries: A review, Renew. Sustain. Energy Rev. 113 (2019) 109254.
[253] Q. Wang, Y. He, J. Shen, Z. Ma, G. Zhong, A unified modeling framework for lithium-ion batteries: An artificial neural network based thermal coupled equivalent circuit model approach, Energy 138 (2017) 118–132.
[254] W. Li, J. Zhang, F. Ringbeck, D. Jöst, et al., Physics-informed neural networks for electrode-level state estimation in lithium-ion batteries, J. Power Sources 506 (2021) 230034.
[255] J. Shi, A. Rivera, D. Wu, Battery health management using physics-informed machine learning: Online degradation modeling and remaining useful life prediction, Mech. Syst. Signal Process. 179 (2022) 109347.
[256] H. Tu, S. Moura, Y. Wang, H. Fang, Integrating physics-based modeling with machine learning for lithium-ion batteries, Appl. Energy 329 (2023) 120289.
[257] B. Sun, J. Pan, Z. Wu, Q. Xia, et al., Adaptive evolution enhanced physics-informed neural networks for time-variant health prognosis of lithium-ion batteries, J. Power Sources 556 (2023) 232432.
[258] Y.A. Yucesan, F.A.C. Viana, A physics-informed neural network for wind turbine main bearing fatigue, Int. J. Progn. Heal. Manag. 11 (1) (2020).
[259] Y.A. Yucesan, F.A.C. Viana, A hybrid physics-informed neural network for main bearing fatigue prognosis under grease quality variation, Mech. Syst. Signal Process. 171 (2022) 108875.
[260] M.J. Mayer, Benefits of physical and machine learning hybridization for photovoltaic power forecasting, Renew. Sustain. Energy Rev. 168 (2022) 112772.
[261] H. Li, Y. Wang, J. Zhu, M. Han, Performance map and operating condition optimization of industrial-size SOFCs using a PINN surrogate model, Int. J. Hydrog. Energy 50 (2024) 1294–1307.
[262] H. Gong, S. Cheng, Z. Chen, Q. Li, Data-enabled physics-informed machine learning for reduced-order modeling digital twin: application to nuclear reactor physics, Nucl. Sci. Eng. 196 (6) (2022) 668–693.
[263] M. Gnatowski, S. Buchaniec, G. Brus, The prediction of the polarization curves of a solid oxide fuel cell anode with an artificial neural network supported numerical simulation, Int. J. Hydrog. Energy 48 (31) (2021) 11823–11830.
[264] L. Ma, J. Tian, T. Zhang, Q. Guo, C. Hu, Accurate and efficient remaining useful life prediction of batteries enabled by physics-informed machine learning, J. Energy Chem. 91 (2024) 512–521.
[265] X. Zhao, K. Shirvan, R.K. Salko, F. Guo, On the prediction of critical heat flux using a physics-informed machine learning-aided framework, Appl. Therm. Eng. 164 (2020) 114540.
[266] M.E.C. Bento, Physics-guided neural network for load margin assessment of power systems, IEEE Trans. Power Syst. (2023).
[267] C.J. Turner, J. Oyekan, L. Stergioulas, D. Griffin, Utilizing industry 4.0 on the construction site: Challenges and opportunities, IEEE Trans. Ind. Inform. 17 (2) (2020) 746–756.
[268] A. Allouhi, Y. El Fouih, T. Kousksou, A. Jamil, et al., Energy consumption and efficiency in buildings: current status and future trends, J. Clean. Prod. 109 (2015) 118–130.
[269] L. De Boeck, S. Verbeke, A. Audenaert, L. De Mesmaeker, Improving the energy performance of residential buildings: A literature review, Renew. Sustain. Energy Rev. 52 (2015) 960–975.
[270] C.E. Kontokosta, C. Tull, A data-driven predictive model of city-scale energy use in buildings, Appl. Energy 197 (2017) 303–317.
[271] H. Wang, Z.J. Zhai, Advances in building simulation and computational techniques: A review between 1987 and 2014, Energy Build. 128 (2016) 319–335.
[272] L. Mei, Q. Wang, Structural optimization in civil engineering: a literature review, Buildings 11 (2) (2021) 66.
[273] T. Hong, Z. Wang, X. Luo, W. Zhang, State-of-the-art on research and applications of machine learning in the building life cycle, Energy Build. 212 (2020) 109831.
[274] M. Azimi, A.D. Eslamlou, G. Pekcan, Data-driven structural health monitoring and damage detection through deep learning: State-of-the-art review, Sensors 20 (10) (2020) 2778.
[275] K. Arendt, M. Jradi, H.R. Shaker, C. Veje, Comparative analysis of white-, gray-and black-box models for thermal simulation of indoor environment: Teaching building case study, in: Building Performance Analysis Conference and SimBuild: Co-Organized By ASHRAE and IBPSA-USA, 2018.
[276] S. Chen, X. Zhou, G. Zhou, C. Fan, P. Ding, Q. Chen, An online physical-based multiple linear regression model for building's hourly cooling load prediction, Energy Build. 254 (2022) 111574.
[277] F. Bünning, B. Huber, A. Schalbetter, A. Aboudonia, et al., Physics-informed linear regression is competitive with two Machine Learning methods in residential building MPC, Appl. Energy 310 (2022) 118491.
[278] G. Gokhale, B. Claessens, C. Develder, Physics informed neural networks for control oriented thermal modeling of buildings, Appl. Energy 314 (2022) 118852.
[279] Y. Chen, Q. Yang, Z. Chen, C. Yan, S. Zeng, M. Dai, Physics-informed neural networks for building thermal modeling and demand response control, Build. Environ. 234 (2023) 110149.
[280] J. Vivian, E. Prataviera, N. Gastaldello, A. Zarrella, A comparison between grey-box models and neural networks for indoor air temperature prediction in buildings, J. Build. Eng. 84 (2024) 108583.
[281] R. Zhang, Y. Liu, H. Sun, Physics-guided convolutional neural network (PhyCNN) for data-driven seismic response modeling, Eng. Struct. 215 (2020) 110704.
[282] Z. Zhang, C. Sun, Structural damage identification via physics-guided machine learning: a methodology integrating pattern recognition with finite element model updating, Struct. Heal. Monit. 20 (4) (2021) 1675–1688.
[283] Y. Li, P. Ni, L. Sun, Y. Xia, Finite element model-informed deep learning for equivalent force estimation and full-field response calculation, Mech. Syst. Signal Process. 206 (2024) 110892.
[284] S. He, W. Zhou, C. Tang, Physics-informed neural networks for settlement analysis of the immersed tunnel of the Hong Kong–Zhuhai–Macau bridge, Int. J. Geomech. 24 (1) (2024) 04023241.
[285] W. Ouyang, G. Li, L. Chen, S. Liu, Machine learning-based soil–structure interaction analysis of laterally loaded piles through physics-informed neural networks, Acta Geotech. (2024) 1–26.