KSME

Springer

Journal of Mechanical Science and Technology 35 (4) 2021

Invited Review Article
DOI 10.1007/s12206-021-0342-5

Keywords:
· Deep neural networks
· Informed deep learning
· Knowledge integration
· Knowledge representation
· Physics-informed
· Taxonomy
· Dynamical system

Correspondence to:
Seungchul Lee
seunglee@postech.ac.kr

Citation:
Kim, S. W., Kim, I., Lee, J., Lee, S. (2021). Knowledge Integration into deep learning in dynamical systems: an overview and taxonomy. Journal of Mechanical Science and Technology 35 (4) (2021) 1331~1342. http://doi.org/10.1007/s12206-021-0342-5

Received February 3rd, 2021
Revised February 15th, 2021
Accepted February 19th, 2021

† Recommended by Editor
No-cheol Park

Knowledge Integration into deep learning in dynamical systems: an overview and taxonomy

Sung Wook Kim[1], Iljeok Kim[1], Jonghwan Lee[1] and Seungchul Lee[1,2,3]

[1]Department of Mechanical Engineering, Pohang University of Science and Technology, 77 Cheongam-ro, Pohang, Korea, [2]Graduate School of Artificial Intelligence, Pohang University of Science and Technology, 77 Cheongam-ro, Pohang, Korea, [3]Institute of Convergence Research and Education in Advanced Technology, Yonsei University, 50 Yonsei-ro, Seoul, Korea

Abstract Despite the sudden rise of AI, it still leaves a question mark to many newcomers on its widespread adoption as it exhibits a lack of robustness and interpretability. For instance, the insufficient amount of training data usually hinders its performance due to the lack of generalization, and the black box nature of deep neural networks does not allow for a precise explanation behind its mechanism preventing a new scientific discovery. Such limitations have led to the development of several branches of deep learning one of which include physics-informed neural networks that will be covered in the rest of this paper. In this overview, we defined the general concept of informed deep learning followed by an extensive literature survey in the field of dynamical systems. We hope to make a contribution to our mechanical engineering community by conveying knowledge and insights on this emerging field of study through this survey paper.

1. Introduction

Deep learning has recently become the epitome of emerging research studies stimulating its widespread adoption in nearly every scientific and engineering areas. Due to its inherent nature to capture high non-linearity between input variables and target variables for various types of data (e.g. image, text, sound, etc.), it is rapidly becoming a near universal tool for many research studies. From the perspective of engineers at manufacturing sites, the best part of using deep learning is that it is not mandatory for users to know about the physical meanings of the parameters of the system of interest, and nor are they required to understand the mechanism behind it. This is important because often times the combination of input parameters such as control settings (e.g. speed, pressure, etc.) and change in physical quantities of conditions (e.g. stress, tensile force, etc.) over time provide an unexpected output.

Although the conventional deep learning approaches, usually regarded as the data-driven methods, have shown promising results in diverse domains, they still present limitations in several aspects that are left to be resolved. For instance, the insufficient amount of training data is detrimental for the performance of a deep learning model, and it is generally considered the first and foremost. To make the matter even worse, there is no strict rule-of-thumb for the minimum amount of data that is necessary as it depends largely on the type of task that needs to be solved. Furthermore, deep learning approaches are solely data-driven in its nature for discovering the hidden patterns or mechanisms and thereby disobey the physical constraints or governing laws occasionally. Apart from those mentioned, there are a few other disadvantages such as the lack of interpretability and a relatively long training time. To overcome these challenges, scientists and engineers have begun to create additional features in the traditional deep learning pipeline. As such, several branches in the field of deep learning have emerged, one of which include the physics-informed deep learning.

Nomenclatures regarding the integration of prior knowledge with deep learning are not unified

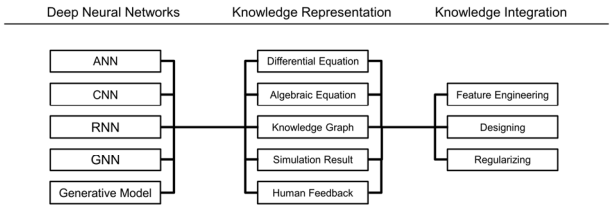

Fig. 1. Taxonomy of informed deep learning.

across previous literature. Some of the commonly used expressions include 'physics-informed', 'physics-based', 'physics-guided' and 'theory-guided'. Therefore, for clarity, we used 'informed deep learning' in this paper to denote the knowledge integration with deep learning. The rest of this study is broken down as follows. In Sec. 2, the general concept of informed deep learning is introduced and explained in detail. Sec. 3 discusses in depth on how the surveyed literature falls into any of the categories defined in Fig. 1. The classification of the surveyed literature is summarized in Table 1. Lastly, Sec. 4 ends this overview with a summary and an insight on possible future work and suggestions regarding the direction of informed deep learning especially in the field of dynamical systems.

2. Concept of informed deep learning

In this section, the general concept of the workflow in informed deep learning is explained first to give the readers a good idea on how it differs from the learning route of the traditional deep learning. Next, the three different ways of knowledge integration presented in the last part of the taxonomy in Fig. 1 are explained in detail. It is noteworthy that a similar concept of knowledge integration was introduced by Reichstein et al. [1].

2.1 General workflow of knowledge integration in deep learning

Prior knowledge refers to any type of knowledge about a task, data, physical model of interest, etc. One of the essential characteristics of prior knowledge is that it is independent of learning algorithms. The type of knowledge is diverse and a part of those in dynamical systems we encountered during the literature survey are differential equation, algebraic equation, knowledge graph, simulation result and human feedback. The definitions for each term are omitted because they are intuitive and can easily be found in other literature sources. This survey focuses primarily on where and how the prior knowledge was utilized in the commonly referred baseline deep neural networks.

Fig. 2 illustrates the workflows in both conventional and informed deep learning pipelines. The main difference between the conventional and the informed deep learning throughout the entire pipeline is that prior knowledge is explicitly used in one or more of the steps inside the 'integration point'. Although the notion of the 'integration point' is similar to what had been

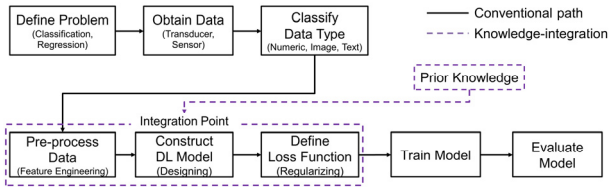

Fig. 2. Workflows in both conventional and informed deep learning pipelines, which show how the route changes from that of the conventional deep learning if prior knowledge is integrated in the pipeline. If following the 'Knowledge-integration' pipeline, the prior knowledge will be taken account in one or more of the steps inside the 'integration point'.

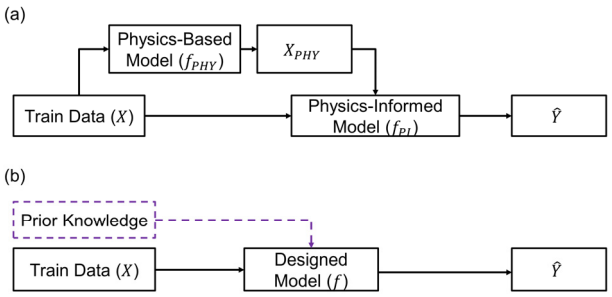

Fig. 3. Process of (a) feature engineering; (b) designing and regularizing as part of knowledge integration.

discussed by Rueden et al. [2, 3], we managed to simplify the four types of ML pipeline into three (i.e. feature engineering, designing, and regularizing) which we thought were more practical and intuitive for the kind of taxonomy shown in Fig. 1.

2.2 Prior knowledge integration

The following sections define steps and the scope of process needed for a method to be categorized as one or more of them.

Feature engineering: Feature engineering in the context of informed deep learning is meant by obtaining supplementary attributes using physics-based model (e.g. simulation, governing equations, etc.) only to have them concatenated with train data, or processing train data with domain-specific functions (e.g. Fourier transform). Despite its restricted usage, feature engineering was found to be a one of the most common ways of integrating prior knowledge to a deep learning pipeline. Concatenating simulation results to the original train data was the most commonly, yet very powerful, encountered method during the literature survey. Fig. 3(a) illustrates the process of feature engineering as part of knowledge integration.

Designing: Designing refers to building a network architecture that reflects the physical knowledge of a system or the type of data. Unlike feature engineering, designing was encountered in a various form. Common cases include constructing a task-specific autoencoder and a recurrent cell, and designing a physics-induced convolution filter. The prior knowledge was usually in the form of a differential equation and a common sense of the domain. Fig. 3(b) illustrates the workflow in which the prior knowledge is shown in a dotted line because it is not physically but indirectly integrated with the designed model

within the scheme.

Regularizing: Regularizing was often coined as physics-guided learning in many literature [2-6]. The main focus of regularizing is to constrain the model objective function in the context of optimization so that the training converges faster and to a certain degree. This form of knowledge integration was found in all types of deep neural networks due to its ease of use. Most of the time, the constraint terms were added to the standard data-driven loss term. The associated workflow is the same as that of designing (Fig. 3(b)).

3. Knowledge integration in deep neural networks

In this section, we provide related studies found in the literature survey. The methods of the studies is focused and investigated for their categorization in the taxonomy shown in Fig. 1. All methods introduced in this section is structured according to the type of deep neural network because we have learned that the way knowledge representations are integrated into the network is largely determined by the type of architecture. For each network, a detailed explanation on how a specific type of knowledge is integrated is given. Then, we classify the proposed method based on the observed paths as shown in Fig. 5.

3.1 Artificial neural networks (ANNs)

ANN was the most frequently used network that we encountered during the literature survey. We assume that it is because of its ease of usage when making modifications to the network for informing various types of knowledge representations. ANN typically consists of an input layer, hidden layers and an output layer. The number of hidden layers, input nodes and output nodes can be easily modified, making it easily applicable to various data types such as sensor data, sound data, image data, and so on.

Zhang et al. [7] presented a way to transfer knowledge learned from one operating condition, hereby referred to as 'simulation result', to model the bearing fault diagnosis in another operating condition. The 'simulation result' is acquired by first constructing an ANN and training it using abundant source data. The learned parameters of the ANN are passed directly to another ANN but with a different output layer due to the target data having different target labels. This process is illustrated in Fig. 4. This way of knowledge transfer is called transfer learning and it is widely used in the world of deep learning for improving the performance in a case where the current task seems challenging to handle with only small amount of target data. It should be noted that the source data and the target data are from different but similar distribution due to different operating conditions. Suppose θ_s and θ_t are parameters in source and target task. They are represented as follows:

$$\theta_s = \theta_0 + \theta_1 \tag{1}$$

$$\theta_t = \theta_0 + \theta_2 \tag{2}$$

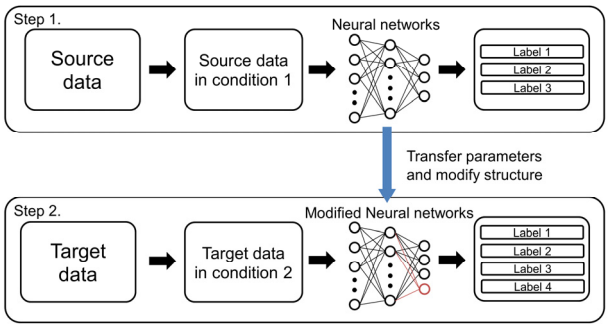

Fig. 4. Process of transfer learning.

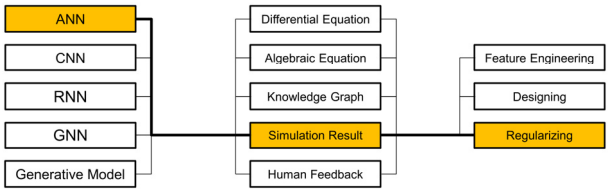

Fig. 5. Taxonomy path for 'ANN – simulation result – regularizing'.

The aim of transfer learning is to find and utilize the common parts θ_0 of the parameters when training the model in target task. As such, the prior knowledge (common parts of the parameters) learned from the source data is used to get a better understanding of the target data.

Karpatne et al. [3] proposed physics-guided neural network (PGNN), a more intuitive method to transfer pre-existing scientific knowledge by using a physics-based simulation result as an additional input to a physics-guided model. A major drawback of data-driven approaches is that the models are solely dependent on the available train data which is often limited in numerous scientific problems. Therefore, the model has a chance to learn spurious relationships that look good only on the train and test sets. Moreover, its predictions may violate the physical laws of the real world due to error associated with the train data. On the other hand, a physics-based model is based on scientific principles such that it enables to explain the relationship between input and output variables. In this case, though the predictions are interpretable, such a model often suffers from inaccurate results due to simplified or missing physics. Assuming that combining the two distinct approaches should resolve their complementary deficiencies and leverage information in both physics and data, a hybrid model f_{HPD} is used with a physics-based loss function L_{Total}:

$$f_{HPD} : \mathrm{X} = \left[D,\ Y_{PHY} \right] \rightarrow Y \tag{3}$$

$$L_{Total} = L\left(\hat{Y}, Y\right) + \lambda R(f) + \lambda_{PHY} L_{PHY}\left(\hat{Y}\right), \tag{4}$$

where D, Y_{PHY}, $L\left(\hat{Y}, Y\right)$, $R(f)$, λ and $L_{PHY}\left(\hat{Y}\right)$ represent input features, simulation output, empirical error, structural error, relative importance and physical inconsistency

error, respectively. Similar studies that use simulation result for a richer representation of train data are presented in Refs. [8-11]. To obtain the simulation output Y_{PHY}, a simulation model [12] was used. Whereas the empirical error and the structural error are the typical loss terms of a purely data-driven model, the physical inconsistency error is specific to the proposed model. The error can be expressed as a rectified linear unit function (ReLU) because the density prediction at time step $t+1$ is less than or equal to the prediction at the previous time step t by the law of physics. The study demonstrated that with the help of the additional physics-based loss term, the generalization performance has been improved even when the train data is small and not fully representative. This flow of work corresponds to the 'ANN – simulation result – regularizing' pipeline shown in Fig. 5. Similar studies but with the designing approach in knowledge integration are presented by Refs. [13, 14].

Yuan et al. [15] presented a similar approach with regard to knowledge integration but with a different type of prior knowledge. The key to successful data-driven structural health monitoring is to acquire as much data associated with diverse damage scenarios as possible. However, it remains a challenge in most cases. As an alternative, physics-informed neural network is presented as a potential remedy in which a partial differential equation modelling the structural behavior of a beam (Fig. 6) is informed in the learning process. In this study, the data shortage scenario is first established by distributing sensors sparsely across a beam, and the reconstruction of the full displacement field is demonstrated using the limited data and the proposed physics-informed model. The suggested loss function used for the task is as follows:

$$\mathcal{L} = MSE_D + MSE_{B1} + MSE_G + MSE_{B2} \tag{5}$$

The loss function can be broken down largely into four parts. The first term MSE_D denotes the purely data-driven loss. The second term MSE_{B1} is for the Dirichlet boundary conditions and the third term MSE_G represent the governing equation. The last term MSE_{B2} plays the same role as the second term but for the higher order boundary conditions. Compared to Ref. [3], the difference is that instead of a loss term that reflects a mere physical property, the entire governing equation (partial differential equation) is embedded to guide the learning. This study shows that embedding the known physics (Euler-Bernoulli beam theory) improves the reconstruction of the full displacement field especially when compared to a purely data-driven model. This flow of work corresponds to the 'ANN –differential equation – regularizing' pipeline in Fig. 1. Studies with the same flow of work are presented in Refs. [16, 17] while studies with algebraic equations as prior knowledge are introduced by Refs. [18, 19]. With ANN as the baseline model, algebraic equations were also used in Refs. [20-22] while human feedback was concerned in Refs. [23-26].

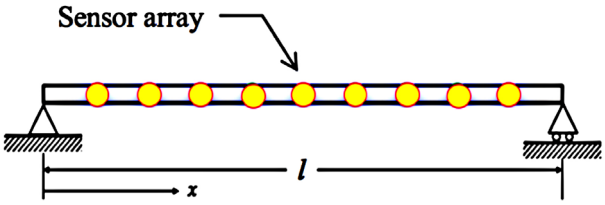

Fig. 6. The structure of a beam with sensor array placed on the beam [15].

3.2 Convolutional neural networks (CNNs)

CNN is a type of ANN but with different body structure composed of convolution layers. This network is most widely used in case where the training data is in the form of image, but numerical data can be also used for training if 1D CNN is the baseline model. The notable feature of CNN is the presence of a convolution filter, in other words, a kernel that typically has a shape of 3 by 3, 5 by 5 and so on. This square-shaped filter is convolved with the input pixels, scanning every squares defined by the user-defined filter size and stride. Since the same filter is used to convolve the input pixels, the number of hyperparameters can be kept small. This filter along with the convolution operation is what makes CNN so unique and powerful with a lower risk of overfitting in the training process.

The traditional data-driven methods using CNN normally trains these filters unless a prior knowledge of the filter is known in advance. For example, when detecting a vertical edge in an input image, the most probable filter would also have a similar looking vertical edge in itself. This is because the convolution operation is the process of finding the similarity between the input and the filter. Unfortunately, in most of the real-world applications, the input data is not in the form of what the user can recognize (e.g. a random matrix), so it is difficult to define the filter matrix at the beginning. However, some scientists and engineers have come up with a way to go around this issue. Sadoughi et al. [27] showed that designing a convolution filter based on a prior knowledge of rotational speed and fault characteristic frequencies of a bearing can greatly enhance the overall performance of CNN. For the fault diagnosis of multiple bearings, the study implements the so-called physics-based convolutional neural network (PCNN) in which one of its layers goes through a physics-based convolution. In particular, the convolution is done with a kernel having a series of reference signals that have been generated by using the following simulation model:

$$\varnothing(k) = a_0 \sum_{s=0}^{+\infty} \left[\chi\left(k - s / f_0\right) \cdot e^{-\xi(k-s/f_0)} \right], \tag{6}$$

where k is the time index and is determined by the rotational speed. f_0 is the fault characteristic frequency of the defective bearings. This flow of work corresponds to the 'CNN – algebraic equation – designing pipeline in Fig. 1.

Zhang et al. [4] presented a physics-guided convolutional neural network (PhyCNN) for seismic response modeling. Re-

cently, the data-driven approach to modelling structural response excited by natural hazards has gained substantial attention in the associated community. The existing conventional methods mostly focus on extracting domain-specific features and using them for fitting models such as the state-space model [28, 29]. Such an updating procedure requires the use of finite element analysis that is computationally heavy in case of high-fidelity models. In this study, as an alternative, a machine learning approach which has proven to be a powerful modeling tool and approximator [30, 31] is used but in a physics-oriented manner. In detail, PhyCNN contains an additional loss term referred to as 'physical loss'. The physical law that is encoded is in fact, the equation of motion that models the dynamic system of interest. Refs. [32, 33] also show the use of regularization terms by adding physical constraints. The author claims that the reason behind using a CNN instead of other deep learning models is because it generally performs better in terms of extracting features from raw signals. Therefore, several convolution layer is constructed in addition to a dropout layer after each convolution layer for reducing the overfitting issue [34]. The study suggests that by using the proposed model as a surrogate model for response prediction, issues regarding data scarcity, physical consistency and overfitting could be alleviated or solved. On the other hand, Refs. [35-38] show the usage of simulation output as input to CNNs and prove that it is an effective and easy way to incorporate prior knowledge. Similarly, Refs. [39-42] are examples of incorporating human feedback at the design stage of CNNs.

3.3 Recurrent neural networks (RNNs)

RNN is a class of neural networks suitable for modelling time-series data. This is due to the recursive structure of its hidden layers that do not only provide outputs at a certain time but also pass hidden states to the next hidden layers. Depending on the type of RNN, the outputs at each time step can also passed on. Often times, it is simply considered as multiple copies of a normal ANN, each passing a message to its successor. Such modification allows it to model data at every single time step and the relations among them. RNNs can have different structures depending on the type of task to be solved. RNNs can structurally be one-to-one, one-to-many, many-to-one and many-to-many. One representative case of where many-to-many RNN structure is most widely used is machine translation. For example, if the encoder part of the network receives words in English, the decoder part of the network should output corresponding translated words in French.

In dynamical systems, most of the governing equations model the behavior of a system over time. As such, they are often expressed as differential equations. Although the governing equations may be inaccurate, this implies that they can be a great source of prior knowledge. Nascimento et al. [43] introduced the use of physics-informed recurrent neural network based on cumulative damage modelling. A cumulative damage model [44, 45] simply estimates a_t, damage at time t by adding damage at the previous time step to Δa_t, damage increment.

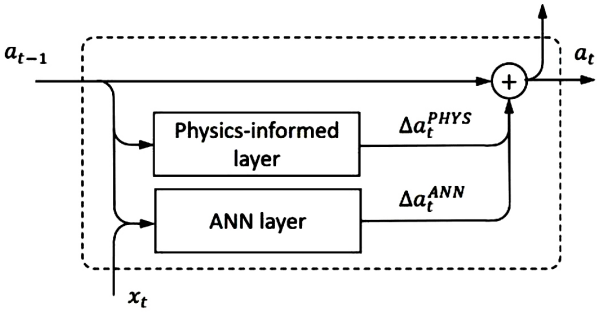

Fig. 7. Hybrid recurrent network cell with an ANN layer for bias estimation [46].

$$a_t = a_{t-1} + \Delta a_t \tag{7}$$

This formulation is actually similar to the formulation of a single RNN cell where the hidden state of the previous time step is given as input to the cell and is passed together with the current observable variables to a perceptron with a sigmoid or tangent activation function to output the hidden state of the current time step. Henceforth, the cumulative damage cell can replace the traditional RNN cell throughout the entire RNN scheme. The modelling of the damage increment is done by having two distinct layers one of which represents a data-driven model while the other does a physics-based one (e.g. Paris law in corrosion-fatigue propagation). Building on top of this work, Dourado et al. [46] also presented physics-informed recurrent neural network for corrosion-fatigue prognosis, in which the crack growth model (Walker model) is coupled with a data-driven layer to model the relatively less understood corrosion-induced damage accumulation. Fig. 7 illustrates the hybrid recurrent network cell for the bias estimation caused by corrosion. This flow of work corresponds to the 'RNN – differential equation – designing' pipeline in Fig. 1. Studies with the same flow of work are presented in Refs. [47-49].

Yu et al. [2] used a similar approach but for structural dynamics simulation. It is stated that the prediction of dynamical response is significant for tasks such as risk assessments and topology optimization. The idea is to replace the traditional way of predicting the dynamical response based on physics-based models and finite element analysis, which is often computationally expensive, with the proposed physics-guided machine learning model based on RNN. In the proposed model, two distinct layers, physics-based layers and data-driven layers are placed in order as shown in Fig. 8. The main difference here is that the physics-based layer is comprised of three of the so-called 'residual blocks' that encode the underlying physics and compute a residual value that reflects the physical consistency. A domain-specific residual function is used for the application:

$$r_{n+1} = x_{n+1}^{pred} - x_n^{pred} - \Delta t\left[Ax_{n+1}^{pred} + Bz\left(t_n + \Delta t\right)\right], \tag{8}$$

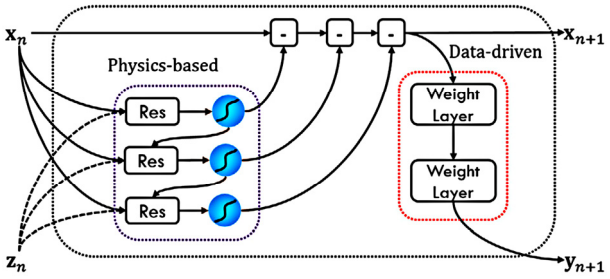

Fig. 8. Proposed physics-guided RNN cell [2].

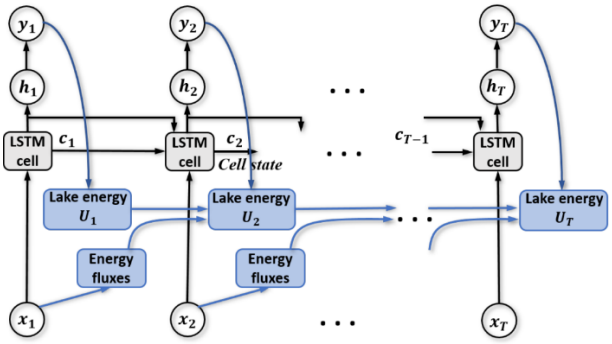

Fig. 9. The recurrent flow of the PGRNN [5].

where x_{n+1}^{pred} and x_n^{pred} are predicted structural responses, Δt is time step, $z(t)$ denotes the system excitation input vector, and A and B represent state matrix and input matrix of the system, respectively.

There are three residual blocks inside the physics-based layers because there needs to be one for each of the equations of motion for a three degree-of-freedom (DOF) spring-dashpot system. Although the intention is to minimize the residual function and zero it for the ideal training in the context of deep residual RNN [50], this is generally hindered due to the system state being directly unobservable which necessitates an additional approximation of relationship between the hidden state and an observable output denoted as y. Henceforth, the data-driven part handled by two MLP layers is introduced right after the residual blocks to conduct the approximation for the physical relationship that is unknown. The study demonstrates that the proposed model predicts the state response very close to the true state response under various circumstances.

Jia et al. [5] presented a case study on predicting lake temperature using a physics-guided RNN (PGRNN) which is not a modified version of the RNN cell like the earlier cases. The model is simply an ordinary LSTM (an extension to RNN) that is constrained during optimization by having two additional physics-induced loss terms. The recurrent model structure in Fig. 9 illustrates the two distinct parallel flows of the proposed model. The first flow is a standard RNN flow which captures the temporal dependency of the given data. The second flow is an energy flow that keeps track of the variation of energy balance over time.

In the context of lake temperature modeling, the main factor of temperature change is known to be heat flux. Heat flux is an energy, and the law of energy conservation must hold. Therefore, the change in energy flux should be equal to the total net amount of incoming and outgoing heat fluxes (terrestrial longwave radiation, short-wave radiation, back radiation, sensible heat fluxes, and latent evaporative heat fluxes) of the system.

$$\mathcal{L} = \mathcal{L}_{RNN} + \lambda_{EC}\mathcal{L}_{EC} + \lambda_{DC}\mathcal{L}_{DC}, \tag{9}$$

$$\mathcal{L}_{EC} \approx \sum ReLU\left(\left|\Delta U_t - \mathcal{F}\right| - \tau_{EC}\right) \tag{10}$$

$$\mathcal{L}_{DC} \approx \sum_t \sum_d ReLU\left(\rho_{d,t} - \rho_{d+1,t}\right) \tag{11}$$

Eq. (9) represents the loss function of the PGRNN. λ_{EC} and λ_{DC} denote relative importance of the corresponding terms. The first term $\mathcal{L}_{RNN}$ is a standard loss function of mean squared error. The second $\mathcal{L}_{EC}$ and third terms $\mathcal{L}_{DC}$ represent the loss of energy conservation and the density-depth constraint, and they are defined by Eqs. (10) and (11), respectively. Eq. (10) is a mathematical formulation of the law of energy conservation. Similarly, Eq. (11) describes the physical constraint that water density, ρ must not get smaller as it gets deeper down in a lake. d and t are depth and time, respectively. In Eq. (10), the ReLU function penalizes cases where the input value is larger than zero. As such, the model is optimized such that the change in energy flux, ΔU_t is the nearly the same as the net amount of in-fluxes and out-fluxes, $\mathcal{F}$. The difference between the two can be allowed by as much as τ_{EC}, which is a pre-determined threshold value.

As for another example of a physics-based RNN, Yuan et al. [15] constructed a CNN-RNN model based on two pieces of domain knowledge about wave propagation that are Huygens' principle and time-reversal technique for source localization in the impact diagnosis of a structure. In accordance with Huygens' principle, multiple time frames of feature vectors extracted from the corresponding wave fields by CNN were used in a sequence. Time-reversal is a technique which enables to discover the source of impact by analyzing the input wave signal in a reverse direction. This forms the basis for constructing a many-to-one RNN model consisting of multiple time frame inputs of feature vectors and a temporal decoder that provides the impact location.

3.4 Graph neural networks (GNNs)

All the deep learning models introduced until now deal with data that have a Euclidean structure. Such data type is structurally regular allowing the data to be fed to the network in an evenly spatial and timely manner. Examples of such data include image and text. However, some data cannot be structured the same way, necessitating for another method to formulate an irregular structure. As an alternative, the notion of graph was introduced and it is widely being studied in the deep learning community as of now.

In computer science, a graph generally refers to a data type

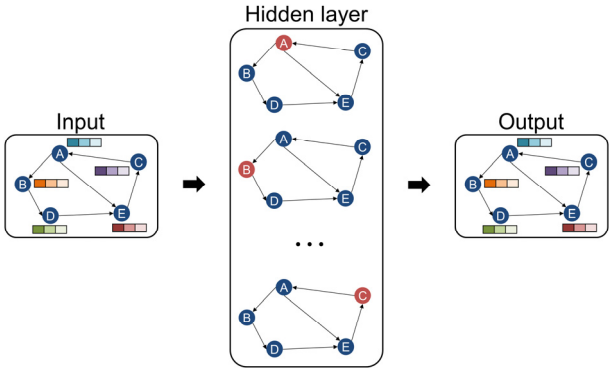

Fig. 10. General flow of a graph convolutional network for node classification. In the hidden layer, every node of the input graph is updated one by one. Since it is a directed graph, the red colored node is affected only by the neighboring nodes that direct toward it.

comprised of two components, nodes and edges. Nodes also called as vertices encode elements while edges encode relations between the elements. Each node is represented by a vector (e.g. a distributed vector representation) which contains meaningful information about the corresponding element of the physical system. Edges can be either directed or undirected, and they can have weights that specify the relative importance of relations between nodes. Such graph structures are frequently used for modeling situations where multiple elements are connected in one way or another and thus one may affect not only the nearby elements but all the rest due to the connections (e.g. a distributed circuit, a smart grid, water supply networks, social networks, etc.). It should be noted that the topology of graph structure itself is an essential source of information if one hopes to use it as training data [51, 52].

The main issue with using graph data is that it is incompatible with the existing deep learning models because the topology should be maintained when it is fed to the models. This problem has led to the advent of GNNs. A major variant of GNN, graph convolutional network (GCN) [53] uses graph Fourier transform to extract features from graphs. Fig. 10 illustrates how the output representation of a graph structure is computed through a GCN. GCN uses graph filters to extract features that are equivalent to the channels within CNNs. The graph filter is a diagonal matrix of which the diagonal terms are also filters parametrized by the eigenvalues of a graph Laplacian matrix, L. L is defined as the difference between the weighted adjacency matrix and degree matrix of a graph. The output layer is chosen depending on the task of the model. For node classification, a fully connected layer is the popular choice.

In line with the development of GNNs, the issue of integrating prior knowledge with GNNs has also attracted several researchers over the past few years. Seo et al. [54] developed a methodology to model a climate dynamical system by incorporating implicit physics knowledge in latent space. The suggested network is composed of an encoder and a decoder. The encoder receives a set of attributes of nodes (v), edges (e), 3-cliques (c), and a whole graph (u) as inputs which are then transformed into latent spaces. After that, the encoded graph is updated multiple times until the minimum requirement is met. It should be noted that the updated graph represents the state of the graph. For this particular domain of climate modeling, the user knows that the observations should possess a diffusive property. For this reason, the diffusion equation is given as an additional constraint to the total objective function. Similarly, Seo et al. [55] introduced the concept of spatial difference layer that adds an additional constraint to the total loss function.

Zhang et al. [56] proposed a novel graph neural network named Circuit-GNN that leverages a lot of domain knowledge for the simulation of distributed circuit design. One of its great contributions is placed on how it addresses the incompatibility of models trained on different circuit templates with varying numbers of resonators by applying a single GNN that takes the resonators in each circuit as nodes in a graph, and their electromagnetic coupling as edges between the nodes. By doing so, the prior knowledge of the circuit is transformed into a form of structured data called knowledge graph that can be trained by GNNs. This solves the issue of the requirement of training multiple separate models per template. At the same time, the aim is to obtain an optimized circuit design given a particular electromagnetic specification (e.g. transfer function), which is known as an inverse problem. To solve it, the author focuses on the differentiable nature of the model. Knowing that the only cause of an invalid circuit is the change in resonator center positions, a few rules are applied to make a constraint in each optimization step. This is to prevent the collision of two or more resonators, a case that does not belong to the valid solution space. Through this approach, the optimal point could be reached much faster and accurately. Other works that exploit knowledge graphs for designing graph neural network architecture are presented in Refs. [57-61].

3.5 Generative models

So far, the deep neural networks that have been introduced fall in the category of discriminative models because they predict the target variable while observing the input variables. On the other hand, a generative model refers to a model of the conditional probability of the observable variable X when the target Y is given. Mathematically, it can be symbolized as follows: $P(X \mid Y = y)$. Simply put, the user is interested in the generation of the observable variables by sampling from the learned probability distributions either implicitly or explicitly. In this section, the cases of the two most widely adopted generative models, variational autoencoders (VAEs) [62] and generative adversarial networks (GANs) [63] are reviewed.

Autoencoder is an unsupervised neural network that learns to compress and encode input data into latent space of smaller dimension, and eventually reconstruct the original input data from the learned latent representation. By the bottleneck design of the architecture, it reduces the data dimension. While

Table 1. Classification of previous studies according to the taxonomy of informed deep learning. FE, D and R stand for feature engineering designing and regularizing, respectively.

		Differential equation	Algebraic equation	Knowledge graph	Simulation result	Human feedback
ANN	FE	-	[20]	-	[7-11]	[23, 24]
	D	-	[21, 22]	-	[13, 14]	[23, 25, 26]
	R	[15-17]	[18, 19]	-	[3]	-
CNN	FE	-	-	-	[35]	-
	D	-	[27]	-	[36]	[15, 39-42]
	R	[4, 32, 33]	-	-	[36-38]	-
RNN	FE	-	-	-	[35]	-
	D	[2, 46-49]	[43]	-	-	[15]
	R	[32]	[5]	-	-	-
GNN	FE	-	-	-	-	-
	D	[57]	[57]	[56-61]	[57, 60, 61]	[56, 60, 61]
	R	[54, 55]	[69]	[54, 69]	-	-
Generative model	FE	-	[70]	-	[35]	-
	D	[71]	-	-	[72]	-
	R	[32, 64, 65, 67, 68]	-	-	[37, 38]	[6]

the former part of the bottleneck network is called an encoder, the latter part is called a decoder. The names literally suggest the role of each part. Autoencoders are widely used for anomaly detection and the de-noising of input signals. Though many more exists, one of the popular variants of autoencoders is VAE. VAE mainly differs from an ordinary autoencoder such that while an autoencoder maps the input into a fixed vector, VAE does it into a distribution. To elaborate, instead of learning a function representing the data, VAE learns the parameters of a probability distribution representing the data. Though the purposes of both models are the same, VAEs are known to provide more control over how the latent space is modeled which is why they are generally preferred over autoencoders.

Chao et al. [6] introduced the so-called "knowledge induced variational autoencoder with adaptive sampling" (KIL-AdaVAE), an extension to VAE, which enables implicit supervision on learning the latent representation of the healthy conditions for anomaly detection in an open-set scenario. In open-set diagnostics, both the fault detection and the fault segmentation should be addressed. Therefore, a meticulous learning of latent representation is needed to identify the healthy condition and distinguish the faulty conditions. In this study, the prior knowledge which is capitalized on is that there are both healthy data and potential faulty data (unlabeled) in the available train set, and thus the representation learning must be balanced for both classes. First, the author stresses the limitation of an ordinary autoencoder for anomaly detection that its latent representation typically shows an entangled representation of the fault types. Henceforth, the use of VAE for better control over representation learning is suggested. However, to bring about a more disentangled representation, implicit supervision by adding a loss term to the VAE is proposed. This additional loss term is intended to restrict the representation of healthy data and encourages a more distinctive representation of the fault types. The proposed loss is as follows:

$$\mathcal{L}_{KIL-AdaVAE} = \mathcal{L}_{ELBO} - \gamma D_{KL}(q_{\varnothing}(z \mid x) \,||\, p(z))_{S_T} \tag{12}$$

S_T represents the healthy labeled system conditions. The second loss term forces the representation of the healthy data to match the factorized unit Gaussian and it acts as the implicit supervision to the unsupervised learning task of VAE. This flow of work corresponds to the 'generative model – human feedback – regularizing' pipeline in Fig. 1.

Generative adversarial networks (GANs) [63] have recently gained huge popularity in the deep learning community as a baseline generative model. Composed largely of two components that compete with each other, a generator and a discriminator, it runs based on the minimax game. The major advantage of GANs is that since it is an implicit model, the user does not need to explicitly designate the form of the underlying probability distributions for the variables of interest. This is why GANs are known to be very powerful in imitating the distributions. Despite its great performance when trained properly, GANs are also known to suffer from several drawbacks such as mode collapse and train instability caused by gradient vanishing and gradient explosion, which make them difficult to provide the best performance.

Based on the model, Warner et al. [64] demonstrated physics-informed generative adversarial networks (PI-GANs) [65] to inversely estimate the elastic modulus (stiffness) in solid mechanics. In this study, the proposed model encodes physical laws in the form of stochastic partial differential equations (PDEs) onto the generator so that it provides realistic and physically coherent outputs. The focus lies on the estimation of

spatially and randomly varying elastic modulus, $E(x,\omega)$ in the form of probability distribution based on its relation to another, observed quantity (two-dimensional observations of material deformation). The governing equation along with boundary conditions, which models the deformation under load at varying space x and random event ω is known in advance, but the distribution of elastic modulus is unknown. The training dataset is generated by solving the governing equation with the boundary conditions using the finite element method. As for the network architecture, two generators (one for $u(x,\omega)$ and the other for $E(x,\omega)$) are constructed and trained in parallel. There is a single discriminator in the network. The loss functions are represented as follows:

$$\mathcal{L}_G^{PI}\left(\theta_{u,E}\right)=\mathcal{L}_G^{\omega}\left(\theta_u\right)+\mathcal{L}_{PDE}\left(\theta_{u,E}\right)+\mathcal{L}_{BC}\left(\theta_{u,E}\right) \tag{13}$$

$$\mathcal{L}_D^{PI}\left(\theta_{u,E},\varphi\right)=\mathcal{L}_D^{\omega}\left(\theta_{u,E},\varphi\right) \tag{14}$$

The above equations represent the generator loss and the discriminator loss. The networks are parameterized by θ and φ, respectively. The first terms of the equations denote the WGAN with gradient penalty loss that replaced a standard GAN losses because of their power to solve the training stability issue of the vanilla GANs [66]. It should be noted that the physics-induced loss terms are added only to the combined generator loss to impose physical constraints only on the generator. Using the aforementioned methodology, it is reported that the distribution of the elastic modulus similar to the true distribution was generated. This flow of work corresponds to the 'generative model – differential equation – regularizing' pipeline shown down below. Similar studies are shown in Refs. [67, 68].

4. Categorization of previous studies

In this section, we provide a summary of the aforementioned literature by putting them into categories to which each study belongs to. The categories are determined based on the taxonomy in Fig. 1. From the Table 1, it is notable that there has been a substantial amount of research in incorporating various types of knowledge with deep learning. There have been cases where the reported works belong to more than just a single category because of a mixture of different deep neural networks (e.g. coupling CNN with RNN) and more than one type of prior knowledge was used in a single case study (e.g. differential equation and algebraic equation are both used). Numerous slots are missing in the table, implying that the field of study is still in its infancy and more research is encouraged.

5. Conclusions

In this overview, we defined the general concept and the taxonomy of informed deep learning followed by an extensive literature survey in the field of dynamical systems on the knowledge integration in deep learning pipelines. As for the general concept, we first defined the meaning of prior knowledge and identified the types of prior knowledge (differential equation, algebraic equation, knowledge graph, simulation result and human feedback) that we encountered during the literature survey. Then, the conventional deep learning and the informed deep learning are compared by their respective workflows. The taxonomy of informed deep learning which we defined in this overview is largely divided into three conceptual steps: deep neural networks, knowledge representation and knowledge integration. Since the notion of deep neural networks and knowledge representation are relatively well understood, they have been explained briefly while the elements of knowledge integration have been explained one by one with some examples in Sec. 2. Although the taxonomy defined in this paper could have included extra features as in some other literatures, we intended to simplify the taxonomy and the knowledge integration part in order to easily convey the knowledge and vision in the emerging field of informed deep learning to our mechanical engineering community. The application of deep learning in the engineering field is still in its infancy not only because of its solely data-driven nature, but also because there has yet been sufficient amount of research for infusing scientific knowledge into deep learning by the related domain experts. For those mechanical engineers who are still very suspicious about data-driven approaches and deep learning may look into this new area of research and obtain good insights on leveraging the benefits of both physics-based models retrieved from dynamical systems and deep learning.

Acknowledgments

This work was partly supported by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIT) (No. 2020R1A2C1009744), Institute for Information & communications Technology Panning & Evaluation (IITP) grant funded by the Korea government (MSIP) (No. 2019-0-01906, Artificial Intelligence Graduate School Program (POSTECH)), the Institute of Civil Military Technology Cooperation funded by the Defense Acquisition Program Administration and Ministry of Trade, Industry and Energy of Korean government under grant No. 19-CM-GU-01, the Korea Institute of Energy Technology Evaluation and Planning (KETEP) Grant funded by the Korean Government (MOTIE) under Grant 20206610100290.

Nomenclature

θ_s : Source parameters
θ_t : Target parameters
f_{HPD} : Hybrid model
L_{PHY} : Physics-based loss
$\varnothing(k)$: Convolution filter
a_t : Damage at time t
r_{n+1} : Residual at $n+1$ step

References

[1] M. Reichstein, G. Camps-Valls, B. Stevens, M. Jung, J. Denzler and N. Carvalhais, Deep learning and process understanding for data-driven Earth system science, *Nature*, 566 (7743) (2019) 195-204.

[2] Y. Yu, H. Yao and Y. Liu, Structural dynamics simulation using a novel physics-guided machine learning method, *Engineering Applications of Artificial Intelligence*, 96 (2020) 103947.

[3] A. Karpatne, W. Watkins, J. Read and V. Kumar, Physics-guided neural networks (pgnn): an application in lake temperature modeling, *arXiv preprint arXiv:1710.11431* (2017).

[4] R. Zhang, Y. Liu and H. Sun, Physics-guided convolutional neural network (PhyCNN) for data-driven seismic response modeling, *Engineering Structures*, 215 (2020) 110704.

[5] X. Jia, J. Willard, A. Karpatne, J. Read, J. Zwart, M. Steinbach and V. Kumar, Physics guided RNNs for modeling dynamical systems: a case study in simulating lake temperature profiles, *Proceedings of the 2019 SIAM International Conference on Data Mining*, SIAM (2019) 558-566.

[6] M. A. Chao, B. T. Adey and O. Fink, Implicit supervision for fault detection and segmentation of emerging fault types with deep variational autoencoders, *arXiv:1912.12502* (2020).

[7] R. Zhang, H. Tao, L. Wu and Y. Guan, Transfer learning with neural networks for bearing fault diagnosis in changing working conditions, *IEEE Access*, 5 (2017) 14347-14357.

[8] M. A. Chao, C. Kulkarni, K. Goebel and O. Fink, Fusing physics-based and deep learning models for prognostics, *arXiv preprint arXiv:2003.00732* (2020).

[9] M. A. Chao, C. Kulkarni, K. Goebel and O. Fink, Hybrid deep fault detection and isolation: combining deep neural networks and system performance models, *arXiv preprint arXiv:1908.01529* (2019).

[10] J. Pfrommer, C. Zimmerling, J. Liu, L. Kärger, F. Henning and J. Beyerer, Optimisation of manufacturing process parameters using deep neural networks as surrogate models, *Procedia CiRP*, 72 (2018) 426-431.

[11] S. J. Leary, A. Bhaskar and A. J. Keane, A knowledge-based approach to response surface modelling in multifidelity optimization, *Journal of Global Optimization*, 26 (3) (2003) 297-319.

[12] M. Hipsey, L. Bruce and D. Hamilton, *General Lake Model Overview and User Information,* The University of Western Austrailia (2014).

[13] H. S. Kim, M. Koc and J. Ni, A hybrid multi-fidelity approach to the optimal design of warm forming processes using a knowledge-based artificial neural network, *International Journal of Machine Tools and Manufacture*, 47 (2) (2007) 211-222.

[14] F. Wang and Q.-J. Zhang, Knowledge-based neural models for microwave design, *IEEE Transactions on Microwave Theory and Techniques*, 45 (12) (1997) 2333-2343.

[15] F.-G. Yuan, S. A. Zargar, Q. Chen and S. Wang, Machine learning for structural health monitoring: challenges and opportunities, *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2020*, International Society for Optics and Photonics (2020) 1137903.

[16] M. Raissi, P. Perdikaris and G. E. Karniadakis, Physics informed deep learning (part i): data-driven solutions of nonlinear partial differential equations, *arXiv preprint arXiv:1711.10561* (2017).

[17] I. E. Lagaris, A. Likas and D. I. Fotiadis, Artificial neural networks for solving ordinary and partial differential equations, *IEEE Transactions on Neural Networks,* 9 (5) (1998) 987-1000.

[18] S. Greydanus, M. Dzamba and J. Yosinski, Hamiltonian neural networks, *Advances in Neural Information Processing Systems* (2019) 15379-15389.

[19] N. Muralidhar, M. R. Islam, M. Marwah, A. Karpatne and N. Ramakrishnan, Incorporating prior domain knowledge into deep neural networks, *2018 IEEE International Conference on Big Data (Big Data)*, IEEE (2018) 36-45.

[20] R. Swischuk, L. Mainini, B. Peherstorfer and K. Willcox, Projection-based model reduction: formulations for physics-based machine learning, *Computers & Fluids*, 179 (2019) 704-717.

[21] Y. Lu, M. Rajora, P. Zou and S. Y. Liang, Physics-embedded machine learning: case study with electrochemical micro-machining, *Machines*, 5 (1) (2017) 4.

[22] C. Bauckhage, C. Ojeda, J. Schücker, R. Sifa and S. Wrobel, Informed machine learning through functional composition, *LWDA* (2018) 33-37.

[23] J. Ling, A. Kurzawski and J. Templeton, Reynolds averaged turbulence modelling using deep neural networks with embedded invariance, *Journal of Fluid Mechanics*, 807 (2016) 155-166.

[24] C.-S. Huang, S.-L. Hung, C. Wen and T. Tu, A neural network approach for structural identification and diagnosis of a building from seismic response data, *Earthquake Engineering & Structural Dynamics,* 32 (2) (2003) 187-206.

[25] A. Butter, G. Kasieczka, T. Plehn and M. Russell, Deep-learned top tagging with a Lorentz layer, *SciPost Phys*, 5 (28) (2018) 1707.08966.

[26] D. L. Bergman, Symmetry constrained machine learning, *Proceedings of SAI Intelligent Systems Conference*, Springer (2019) 501-512.

[27] M. Sadoughi and C. Hu, Physics-based convolutional neural network for fault diagnosis of rolling element bearings, *IEEE Sensors Journal*, 19 (11) (2019) 4181-4192.

[28] B. Moaveni, J. P. Conte and F. M. Hemez, Uncertainty and sensitivity analysis of damage identification results obtained using finite element model updating, *Computer-Aided Civil and Infrastructure Engineering*, 24 (5) (2009) 320-334.

[29] S. Yousefianmoghadam, I. Behmanesh, A. Stavridis, B. Moaveni, A. Nozari and A. Sacco, System identification and modeling of a dynamically tested and gradually damaged 10-story reinforced concrete building, *Earthquake Engineering & Structural Dynamics*, 47 (1) (2018) 25-47.

[30] K. Hornik, Approximation capabilities of multilayer feedforward networks, *Neural Networks,* 4 (2) (1991) 251-257.

[31] C. Tianping and C. Hong, Approximations of continuous functions by neural networks with application to dynamic system, *IEEE Transition Neural Networks*, 4 (6) (1993) 910-918.

[32] E. de Bezenac, A. Pajot and P. Gallinari, Deep learning for

physical processes: incorporating prior scientific knowledge, *Journal of Statistical Mechanics: Theory and Experiment*, 2019 (12) (2019) 124009.

[33] Y. Zhu, N. Zabaras, P.-S. Koutsourelakis and P. Perdikaris, Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data, *Journal of Computational Physics*, 394 (2019) 56-81.

[34] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, *The Journal of Machine Learning Research,* 15 (1) (2014) 1929-1958.

[35] R. King, O. Hennigh, A. Mohan and M. Chertkov, From deep to physics-informed learning of turbulence: diagnostics, *arXiv preprint arXiv:1810.07785* (2018).

[36] Y. Du, Z. Liu, H. Basevi, A. Leonardis, B. Freeman, J. Tenenbaum and J. Wu, Learning to exploit stability for 3d scene parsing, *Advances in Neural Information Processing Systems*, 31 (2018) 1726-1736.

[37] K.-H. Lee, G. Ros, J. Li and A. Gaidon, Spigan: Privileged adversarial learning from simulation, *arXiv preprint arXiv:1810.03756* (2018).

[38] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang and R. Webb, Learning from simulated and unsupervised images through adversarial training, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017) 2107-2116.

[39] T. Cohen and M. Welling, Group equivariant convolutional networks, *International Conference on Machine Learning* (2016) 2990-2999.

[40] S. Dieleman, J. De Fauw and K. Kavukcuoglu, Exploiting cyclic symmetry in convolutional neural networks, *arXiv preprint arXiv:1602.02660* (2016).

[41] J. Li, Z. Yang, H. Liu and D. Cai, Deep rotation equivariant network, *Neurocomputing*, 290 (2018) 26-33.

[42] D. E. Worrall, S. J. Garbin, D. Turmukhambetov and G. J. Brostow, Harmonic networks: deep translation and rotation equivariance, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017) 5028-5037.

[43] R. G. Nascimento and F. A. Viana, Fleet prognosis with physics-informed recurrent neural networks, *arXiv preprint arXiv:1901.05512* (2019).

[44] A. Fatemi and L. Yang, Cumulative fatigue damage and life prediction theories: a survey of the state of the art for homogeneous materials, *International Journal of Fatigue,* 20 (1) (1998) 9-34.

[45] D. M. Frangopol, M. J. Kallen and J. M. v. Noortwijk, Probabilistic models for life-cycle performance of deteriorating structures: review and future directions, *Progress in Structural Engineering and Materials*, 6 (4) (2004) 197-212.

[46] A. Dourado and F. A. Viana, Physics-informed neural networks for corrosion-fatigue prognosis, *Proceedings of the Annual Conference of the PHM Society* (2019).

[47] A. D. Dourado and F. Viana, Physics-informed neural networks for bias compensation in corrosion-fatigue, *AIAA Scitech 2020 Forum* (2020) 1149.

[48] Y. Long, X. She and S. Mukhopadhyay, HybridNet: integrating model-based and data-driven learning to predict evolution of dynamical systems, *arXiv preprint arXiv:1806.07439* (2018).

[49] M. Lutter, C. Ritter and J. Peters, Deep lagrangian networks: using physics as model prior for deep learning, *arXiv preprint arXiv:1907.04490* (2019).

[50] J. N. Kani and A. H. Elsheikh, DR-RNN: A deep residual recurrent neural network for model reduction, *arXiv preprint arXiv:1709.00939* (2017).

[51] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li and M. Sun, Graph neural networks: a review of methods and applications, *arXiv preprint arXiv:1812.08434* (2018).

[52] A. Ortega, P. Frossard, J. Kovačević, J. M. Moura and P. Vandergheynst, Graph signal processing: overview, challenges, and applications, *Proceedings of the IEEE*, 106 (5) (2018) 808-828.

[53] T. N. Kipf and M. Welling, Semi-supervised classification with graph convolutional networks, *arXiv preprint arXiv:1609.02907* (2016).

[54] S. Seo and Y. Liu, Differentiable physics-informed graph networks, *arXiv preprint arXiv:1902.02950* (2019).

[55] S. Seo, C. Meng and Y. Liu, Physics-aware difference graph networks for sparsely-observed dynamics, *International Conference on Learning Representations* (2019).

[56] G. Zhang, H. He and D. Katabi, Circuit-GNN: graph neural networks for distributed circuit design, *International Conference on Machine Learning* (2019) 7364-7373.

[57] A. Mojallal and S. Lotfifard, Multi-physics graphical model-based fault detection and isolation in wind turbines, *IEEE Transactions on Smart Grid*, 9 (6) (2017) 5599-5612.

[58] H. Khorasgani, A. Hasanzadeh, A. Farahat and C. Gupta, Fault detection and isolation in industrial networks using graph convolutional neural networks, *2019 IEEE International Conference on Prognostics and Health Management (ICPHM)*, IEEE (2019) 1-7.

[59] K. Chen, J. Hu, Y. Zhang, Z. Yu and J. He, Fault location in power distribution systems via deep graph convolutional networks, *IEEE Journal on Selected Areas in Communications*, 38 (1) (2019) 119-131.

[60] Z. Gao, G. Lu and P. Yan, Graph-based change detection for condition monitoring of industrial machinery: an enhanced framework for non-stationary condition signals, *Measurement Science and Technology*, 30 (11) (2019) 115002.

[61] T. Wang, G. Lu, J. Liu and P. Yan, Graph-based change detection for condition monitoring of rotating machines: techniques for graph similarity, *IEEE Transactions on Reliability*, 68 (3) (2018) 1034-1049.

[62] D. P. Kingma and M. Welling, Auto-encoding variational bayes, *arXiv preprint arXiv:1312.6114* (2013).

[63] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, Generative adversarial nets, *Advances in Neural Information Processing Systems* (2014) 2672-2680.

[64] J. E. Warner, J. Cuevas, G. F. Bomarito, P. E. Leser and W. P. Leser, Inverse estimation of elastic modulus using physics-

informed generative adversarial networks, *arXiv preprint arXiv: 2006.05791* (2020).
[65] L. Yang, D. Zhang and G. E. Karniadakis, Physics-informed generative adversarial networks for stochastic differential equations, *SIAM Journal on Scientific Computing*, 42 (1) (2020) A292-A317.
[66] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin and A. C. Courville, Improved training of wasserstein gans, *Advances in Neural Information Processing Systems,* 30 (2017) 5767-5777.
[67] Y. Yang and P. Perdikaris, Physics-informed deep generative models, *arXiv preprint arXiv:1812.03511* (2018).
[68] Y. Yang and P. Perdikaris, Adversarial uncertainty quantification in physics-informed neural networks, *Journal of Computational Physics*, 394 (2019) 136-152.
[69] X. Jia, J. Zwart, J. Sadler, A. Appling, S. Oliver, S. Markstrom, J. Willard, S. Xu, M. Steinbach and J. Read, Physics-guided recurrent graph networks for predicting flow and temperature in river networks, *arXiv preprint arXiv:2009.12575* (2020).
[70] V. H. Nguyen, J. S. Cheng, Y. Yu and V. T. Thai, An architecture of deep learning network based on ensemble empirical mode decomposition in precise identification of bearing vibration signal, *Journal of Mechanical Science and Technology*, 33 (1) (2019) 41-50.
[71] F. de Avila Belbute-Peres, K. Smith, K. Allen, J. Tenenbaum and J. Z. Kolter, End-to-end differentiable physics for learning and control, *Advances in Neural Information Processing Systems* (2018) 7178-7189.
[72] A. S. Qureshi, A. Khan, A. Zameer and A. Usman, Wind power prediction using deep neural network based meta regression and transfer learning, *Applied Soft Computing*, 58 (2017) 742-755.

Sung Wook Kim received a B.S. degree in Mechanical Engineering from Hanyang University, Seoul, South Korea, in 2016. He then received his M.S. degree in Mechanical Engineering from Pohang University of Science and Technology, Pohang, South Korea, in 2018. He is now a Ph.D. candidate at the Industrial AI Lab. of Pohang University of Science and Technology. His research interests include industrial artificial intelligence with mechanical systems, and deep learning for smart manufacturing.

Iljeok Kim received a B.S. degree in Mechanical Engineering from Chungnam National University, Daejeon, South Korea, in 2017. He then received his M.S. degree in Mechanical Engineering from Pohang University of Science and Technology, Pohang, South Korea, in 2020. He is now a Ph.D. student at the Industrial AI Lab. of Pohang University of Science and Technology. His research interests include industrial artificial intelligence with mechanical systems, and deep learning for smart manufacturing.

Jonghwan Lee received a B.S. degree in Mechanical Engineering from Kyungpook National University, Daegu, South Korea, in 2018. He is now a M.S. candidate at the Industrial AI Lab. of Pohang University of Science and Technology. His research interests include industrial artificial intelligence with mechanical systems, and deep learning for smart manufacturing.

Seungchul Lee received a B.S. degree from Seoul National University in 2001. He then received his M.S. and Ph.D. degrees from the University of Michigan, Ann Arbor, USA, in 2008, and 2010, respectively. He was an Assistant Professor with the Ulsan National Institute of Science and Technology, South Korea. He is currently an Assistant Professor at the Department of Mechanical Engineering at Pohang University of Science and Technology in Pohang, South Korea, since 2018. His research interests include industrial artificial intelligence with mechanical systems, deep learning for machine healthcare, and the IoT-based smart manufacturing.