



**HAL**  
open science

# A Review of Machine Learning Methods Applied to Structural Dynamics and Vibroacoustic

Barbara Zaparoli Cunha, Christophe Droz, Stéphane Foulard, Mohamed Ichchou, Abdelmalek Zine

► **To cite this version:**

Barbara Zaparoli Cunha, Christophe Droz, Stéphane Foulard, Mohamed Ichchou, Abdelmalek Zine. A Review of Machine Learning Methods Applied to Structural Dynamics and Vibroacoustic. 2022. hal-03563614v1

**HAL Id: hal-03563614**

**<https://hal.science/hal-03563614v1>**

Preprint submitted on 9 Feb 2022 (v1), last revised 26 Jul 2023 (v3)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A Review of Machine Learning Methods Applied to Structural Dynamics and Vibroacoustic

Barbara Zaparoli Cunha<sup>a,b,\*</sup>, Christophe Droz<sup>c</sup>, Abdel-Malek Zine<sup>d</sup>, Stéphane Foulard<sup>b</sup>, Mohamed Ichchou<sup>a</sup>

<sup>a</sup>Laboratory of Tribology and Dynamics of Systems, Ecole Centrale Lyon, Ecully, France

<sup>b</sup>Compredict GmbH, Darmstadt, Germany

<sup>c</sup>Univ. Gustave Eiffel, Inria, COSYS/SH, I4S team, Rennes, France

<sup>d</sup>Institut Camille Jordan, Ecole Centrale Lyon, Ecully, France

---

## Abstract

The use of Machine Learning (ML) has rapidly spread across several fields, having encountered many applications in Structural Dynamics and Vibroacoustic (SD&V). The increasing capabilities of ML to unveil insights from data, driven by unprecedented data availability, algorithms advances and computational power, enhance decision making, uncertainty handling, patterns recognition and real-time assessments. Three main applications in SD&V have taken advantage of these benefits. In Structural Health Monitoring, ML detection and prognosis lead to safe operation and optimized maintenance schedules. System identification and control design are leveraged by ML techniques in Active Noise Control and Active Vibration Control. Finally, the so-called ML-based surrogate models provide fast alternatives to costly simulations, enabling robust and optimized product design. Despite the many works in the area, they have not been reviewed and analyzed. Therefore, to keep track and understand this ongoing integration of fields, this paper presents a survey of ML applications in SD&V analyses, shedding light on the current state of implementation and emerging opportunities. The main methodologies, advantages, limitations, and recommendations based on scientific knowledge were identified for each of the three applications. Moreover, the paper considers the role of Digital Twins and Physics Guided ML to overcome current challenges and power future research progress. As a result, the survey provides a broad overview of the present landscape of ML applied in SD&V and guides the reader to an advanced understanding of progress and prospects in the field.

*Keywords:* Machine Learning, Structural Health Monitoring, Surrogate Model, Active Vibration Control, Active Noise Control, Digital-Twin, Physics Guided Machine Learning

---

## 1. Introduction

In the current Information Era, unprecedented amount of information is produced, stored, and transformed into actionable knowledge [1]. However, such a large amount of data requires processing and translation abilities beyond human capacity. Machine Learning (ML) algorithms have been a key part of the Big-Data revolution, as they play the role of automatically processing these copious amounts of data to extract patterns and make inferences and predictions based on them. In other terms, digitalization and connectivity provide the data, and ML translates it into meaningful information.

Besides the availability of data, ML progress is powered by constant developments in computing resources and algorithm improvements. Currently, ML is widely present in our daily life, such as in health-care decision making [2], autonomous vehicles [3], economic forecasts [4], detection of fake-news [5], suggestions for consumption of content and goods [6, 7], mastering games [8], image classification and generation [9, 10], translations and speech recognition [11] and other subjects.

ML methods are also permeating the natural sciences [12], not only by overcoming traditional data-driven methods but also by powering or even replacing first-principle models. The use of ML in scientific fields such as biology [13], chemistry [14, 15], physics [16–19] and material science [20, 21] is well developed. The range of ML applications in these domains includes identify-

---

\*Corresponding author at École Centrale de Lyon. 36, Avenue Guy de Collongue, 69134, Ecully, France. E-mail address: barbara.zaparoli-cunha@ec-lyon.fr (B.Z. Cunha)

ing behaviors from measured data, speeding up analyses time, merging data- and domain-based knowledge, finding new materials and components, modeling systems, and discovering governing equations. Given this trend, much has been debated about the pros and cons of using ML in physical science and how it can power research progress in engineering domains such as fluids dynamics [22], seismology [23, 24], thermal transport [25, 26] and energy systems [27].

Recently, many relevant works in structural dynamics and vibroacoustic have used ML in three major application areas: Structural Health Monitoring (SHM) using vibration and noise signals [23, 28–97], Active Noise and Vibration Control [23, 98–145] and vibroacoustic Project Design [84, 146–181].

SHM benefits from the ML advantages of extracting relevant features from big data to detect and classify failures efficiently and make lifetime predictions. In Active Control, ML stands out for identifying light models of the system, since the mechanistic models are currently unknown, incomplete, or high-dimensional. Besides that, a miscellaneous of approaches uses ML to model and optimize the controller design. In Vibroacoustic Design, ML-based surrogates result in fast simulations that enable an optimized and robust design for Noise, Harshness and Vibration (NVH). The ML workflow in these applications should consider the characteristics of the vibration or sound signals under analysis.

As supported by the numerous results cited throughout this article, there are many benefits to employing ML in SD&V problems. However, drawbacks, misuses, and difficulties can also be spotted and show the potential for further advancement in the field. The lack of interpretability and physical basis are the aspects that raise more apprehension in the use of ML in SD&V and other physical sciences. Furthermore, although the wave behavior of SD&V systems encloses frequency information which is well explored in SHM, it also leads to non-monotonic and rough functions behaviors, raising challenges to ML models. Currently, implementations in the industry are limited by the lack of substantial amounts of labeled data required in Deep Learning or by the cost of ML simulations in real-time applications. Another issue still open to debate is reasoning about when the use of ML is justifiable and brings gains in time and precision with an adequate level of confidence. The present paper discuss these issues alongside references and approaches that tried to tackle them, indicating viable solutions.

Therefore, this work focuses on doing an original and extensive review of the main contributions and on the emerging opportunities of ML applied in structural dy-

namics and vibroacoustic. The review provides a comprehensive state-of-the-use and guidelines of ML applications in SHM, Active Control, and Product Design and raises the strengths and weaknesses of ML in each of these fields. The present implementation scenario of each application is presented alongside reasoning about method choices and discussion on the identified research gaps. In-depth theory on ML and vibroacoustic are not part of the scope of this review. In that way, the present work intends to guide vibroacoustic engineers willing to explore ML techniques by providing the current background and the future opportunities of the research field merging ML with SD&V.

At first, Section 2 provides the basis of the main ML methods employed in SD&V literature. Section 3 analyzes the ML workflow in SHM, especially how to prepare vibration and noise signals to increase the ML capabilities to identify patterns, and also introduces relevant works in damage detection, diagnosis, and prognosis. Section 4 reviews the applications of ML in Active Control of noise and vibration, including system identification, reduced-order models, sensor and actuator placement, and controller design. Section 5 focus on the use of surrogate models of SD&V simulations to improve Project Design, with attention to their use in uncertainty propagation, sensitivity analysis, and optimization. Section 6 addresses the trends and perspectives in the field by analyzing how integration, physics guidance, and other aspects can leverage the ML impact and applicability in SD&V research and discussing upcoming opportunities from the integration of these fields.

## 2. Overview of Machine Learning Methods

Machine Learning is an Artificial Intelligent (AI) algorithm which makes an **inference from data and experience without the use of explicit programming**. In a simplified way, for a **dataset**  $X$  containing some sets of inputs  $x$ , a **model**  $m(\omega, x)$  with parameters  $\omega$  is defined to represent the relationships and patterns of the dataset and an assessment criterion called **cost function**  $C(X, m(\omega, x))$  is defined to quantify how well the model represent the dataset. After an optimization that minimizes the cost function in relation to the model parameters  $\omega$ , the optimal model  $m(\omega_{opt}, x)$  is used to infer the relationship of unseen data.

Therefore, three key elements describe an ML model: **representation**, which is the chosen way to model data relations through  $m(\omega, x)$  and establish a hypothesis space of all possible models considered, e.g., decision trees, neural networks, hyperplane representations;

**evaluation**, which determines how to access the model performance employing a cost function  $C(X, m(\omega, x))$ , e.g., accuracy, squared error, K-L divergence; and **optimization**, that is the search method used to minimize the cost function, e.g.: gradient descent, greedy search, quadratic programming [182].

Previously to train an ML model, it is crucial to conduct correctly the dataset **sampling**, as well as the **data preprocessing** stages to make the data suitable and more meaningful to the training stage. During the **training** stage, the optimization algorithm searches for the configuration that better represents the training dataset according to the criteria of the cost function. In other words, the ML model is learning with data. Once the model is trained, it can make **predictions** in new data.

The goal of the ML model is to **generalize** well for new unseen data. In other words, the goal is to have a *prediction model* of the true hidden distribution and not a *fitting model* of a sample of this true distribution (the training set). Therefore, minimizing the cost function during training does not guarantee an adequate predictive model, and the final assessment of the ML model must be based on its prediction performance in the new/unseen data from the *test dataset*.

This leads to a crucial concept in ML, the **bias-variance trade-off**, which stands that the generalization error is a combination of bias, variance, and irreducible errors in supervised learning, as illustrated in Figure 1-a. The bias error measures the level of incorrect hypotheses in the model and decreases with model complexity. The variance error measures the variability of model predictions and typically increases with model complexity. Therefore, a high-bias model oversimplifies the problem, leading to bad predictions in both training and test dataset (underfitting), while a low-bias model performs well in the training dataset but might lead to high-variance error (overfitting). Figure 1-a illustrates this trade-off for a given number of training points, and Figure 1-a shows how complex models with low bias become viable with increasing database size.

Techniques used to control overfitting play a leading role in ML algorithms, setting a good balance between bias and variance. A key aspect is the **hyperparameters selection**, which is done manually by experts reasoning, or automatically via algorithms of search and optimization [183–188]. During hyperparameters selection, the generalization error estimative must use samples kept out of the training dataset, which is usually done through cross-validation or by separating a validation dataset.

**Regularization** techniques are also an important strategy to fight overfitting. The regularization tech-

nique varies depending on the ML method, but the motivation is to penalize model complexity and increase robustness to ill-posed problems. The most common regularization strategy is to add a regularization term in the loss function to account for model complexity.

This ML overview mentions the fundamentals which guide most ML methods. The reader can refer to the rich ML bibliography for in-depth theory and methodology, as in the classic ML textbooks by Bishop [189], Friedman et al. [190] and Goodfellow et al. [183]. For an introduction in ML, the authors recommend the article "A high-bias, low-variance introduction to Machine Learning for physicists" from Mehta et al. [17]. This article presents a brief and comprehensible explanation of the main ML concepts, along with tutorials and *Jupyter notebooks* of popular ML algorithms. The article of Domingos [182] is also a reference for providing valuable expertise in implementing successful ML algorithms.

Usually, the ML methods are classified according to the learning approach as:

- **Supervised Learning:** use labeled output as ground truth during training. It is a **regression** model when the output has continuous values or a **classification** if the output are categories or discrete values.
- **Unsupervised Learning:** no labeled output and no correct answer is provided. The algorithm searches for underlying patterns in the data to assume its structures, through clustering or association.
- **Reinforcement Learning:** the algorithm interacts with an environment, earning points to reinforce successful decisions.

Hereafter, the most relevant ML methods applied in recent literature on structural dynamics and acoustics are outlined. Diversification was an important criterion in the choice of the methods, to enclose different applications and ML categories. In Section 2.1 addresses Neural Networks and Deep Learning methods, since they are certainly the largest field of ML and due to their flexibility, they have supervised, unsupervised and reinforced learning algorithms and handle various data structures. Section 2.2 explores Decision Tree based methods, examples of classical ML algorithms which are extremely popular in supervised learning, in special due to their interpretability and their capabilities when applied with ensemble methods. Section 2.3 is dedicated to Gaussian Processes models, popular due

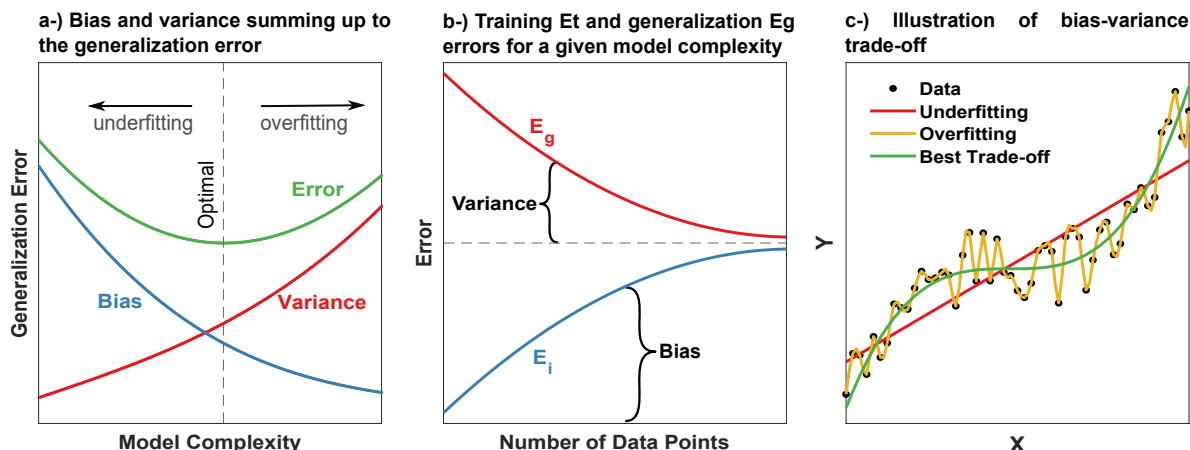


Figure 1: Bias-variance tradeoff on Machine Learning models: (a) Illustration of how bias and variance vary in opposite ways with respect to ML model complexity and how they sum up to the generalization error and, avoiding underfitting and overfitting; (b) Illustration of how the error in the training dataset  $E_t$  is smaller than the true generalization error  $E_g$  and how the prediction accuracy improves with more samples in the dataset; (c) Example of a model that underfits the data, a model that overfits, and a model with appropriate bias-variance tradeoff.

to their probabilistic outputs and mainly used in supervised learning with small database. The K-means algorithm is presented in Section 2.4, as a common example of unsupervised learning. Finally, a brief discussion about reinforcement learning and its class of algorithms called Q-learning is presented in Section 2.5. Certainly, ML is much richer in methods and details than those presented here, but the purpose of this section is just to provide a broad overview of ML methods and to distinguish their advantages and applications in vibroacoustic.

### 2.1. Neural Networks

Neural Networks (NN) or Artificial Neural Networks are a set of ML algorithms inspired in the human brain which can approximate any function, as stated in the Universal Approximation Theorem [191, 192]. Moreover, NN is flexible and modular, so its architecture can be adapted to different purposes in supervised, unsupervised, and reinforced learning. It is by stacking multiple layers in an NN that **Deep Learning** (DL) models are created with improved capabilities of extracting features and learning complex data representations. No wonder, NN is a popular algorithm in all fields of ML, having continually shown impressive results in all kinds of real-life problems.

The most common and general-purpose NN are the **Fully Connected Feed-forward Neural Network** or *Multilayer Perceptron* (MLP) [193], which will be used as a starting point to introduce other NN architectures. As shown in Figure 2 and as the name illustrates, a NN

is a network of artificial neural units. Each neural unit is defined by a nonlinear activation function, as ReLU and sigmoid, which fires an output based on the weighted sum of inputs added to a bias. The outputs from one layer are the inputs for the next one, in a feed-forward procedure. The last layer outputs are the NN predictions, used to evaluate the NN performance. For example, in a supervised problem, the mean squared error between NN predictions and the true values is often used as the NN loss function.

The learning procedure consists of finding the set of weights  $\omega$  and biases  $b$  which minimize the loss. The key to making this optimization viable in big NNs is the **Backpropagation** algorithm, whose rediscovery in the mid-1980s led to the boom in the popularization of NN [194, 195]. Backpropagation allows to efficiently compute the gradient of the loss function with respect to the weights and biases, thanks to the **Automatic Differentiation (AD)** capabilities of NN [196]. Thus, backpropagation enables gradient-based optimization algorithms to be effectively used during training. To reduce memory requirements and speed up the training, the dataset might be divided in *batches*, so that the NN parameters are updated evaluating NN performance at each batch. One training *epoch* has passed when all batches in the dataset have been used to update the NN. The trained NN is a system of algebraic equations which can readily predict new outputs.

NNs account with regularization techniques as *weight regularization* (L1 and L2 regularization), *Dropout*, *Early Stopping*, *Data Augmentation*, *Soft Weight-*

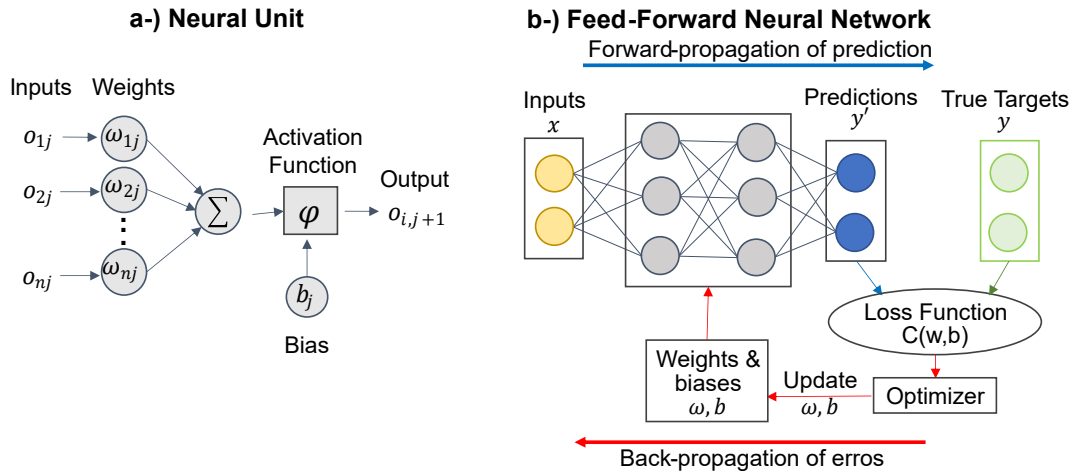


Figure 2: In the neural unit of a NN, the weighted sum of the inputs is added to a bias and goes through a nonlinear activation function, firing the neuron output (a). Supervised training workflow of a Fully Connected Feed-Forward Neural Network with backpropagation of the errors (b).

*Sharing, Batch Normalization and Entropy Regularization* [17, 183, 189, 194]. Besides that, the hyperparameters selection also plays a significant role in NN generalization, since it tunes the architecture aspects and regularization parameters. Basic reasoning for hyperparameters tuning in NN is already well-known [183], but automatized approaches are becoming increasingly popular [197]. In most cases, the use of a validation set to monitor the NN generalization is a standard procedure.

The number of nodes in the first NN layer is equal to the input dimension and the number of nodes in the last layer is equal to the number of output values (regression problem) or classes (classification problem) of the analysis. Usually, in the output layer, a *linear activation function* is used in regression problems with unbounded output, a *softmax activation function* is used for classification problems with multiple mutually exclusive categories and *sigmoid activation function* when you have not mutually exclusive categories. The number of hidden layers, the nodes in each layer and the activation functions in the hidden layers are hyperparameters to be tuned. A crucial point in NN performance is **data scaling**, which makes the data range like the best range for the operation of the activation function. LeCun et al. [198] further discusses practical recommendations for the creation of NN models.

MLP is the most common ML method in the applications covered in this article, due to its versatility and easy implementation. However, there are NN architectures that are appropriate to specific problem structures. Further comments are presented about Convolution Neural Networks and Recurrent Neural Network,

for supervised learning of images and time series, respectively; Auto-encoders for unsupervised learning; and Deep Reinforcement Learning (Section 2.5). Other important NN architectures that are not discussed here are Boltzmann machines, deep belief networks, and generative adversarial networks.

Although the literature on NN is extremely dense and expands fast, there are several references which manage to cover the topic in a didactic way. The book by Nielsen [192] contains comprehensive explanations of the NN main elements, while the classic book by Goodfellow et al. [183] has equally good NN introductions but also covers more detailed and advanced aspects. Mehta et al. [17] presents a gentle and summarized NN guide along with Python code. Implementations guides are available along with dedicated libraries for NN in *Matlab* [199] and in *Python*, with highlights to *Keras* [194] and *PyTorch* [200] libraries.

The authors recommend the survey by Alom et al. [201] to get a broad vision on DL methods, from their basic concepts to state of art algorithms. LeCun et al. [195] and Schmidhuber [202] reviewed Deep Neural Networks (DNN) relevant applications and perspectives. Other important topics in DNN include Adaptive Learning [203], Transfer Learning [204], Online Learning [205] and Probabilistic NN [189, 206].

### 2.1.1. Convolutional Neural Networks

Convolutional Neural Networks (CNN or ConvNet) are NN designed to capture spatial patterns from multiple arrays input by exploiting the **local connectivity** and **translational invariance** characteristics from data. To put it more simply, the CNN architecture considers

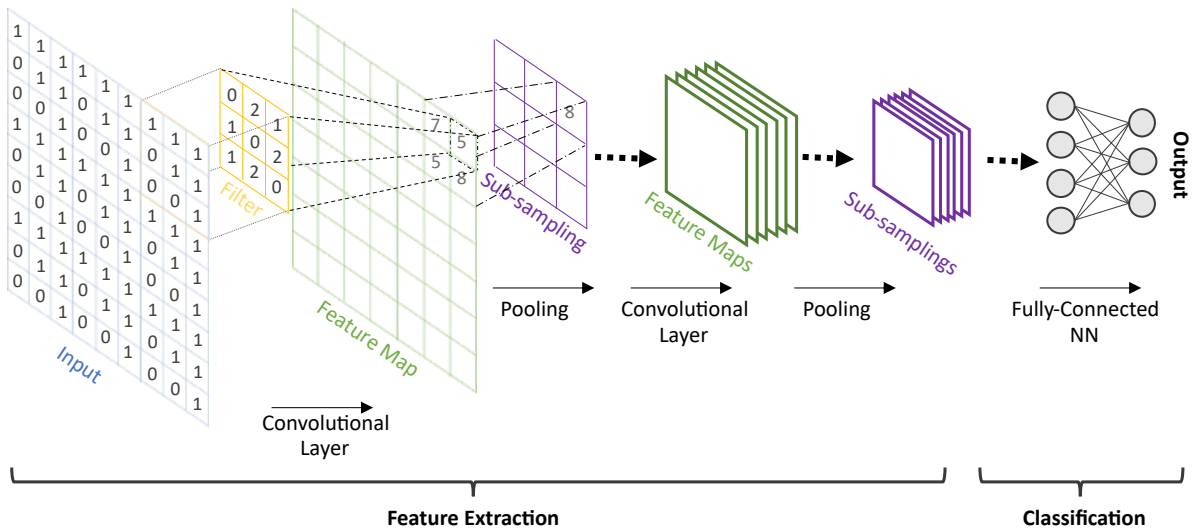


Figure 3: Representation of a Convolutional Neural Network. The CNN performs automatic spatial feature extraction from images by successively applying feature filters that create feature maps (Convolutional layer) and compressing these maps (Pooling layer). Based on the final feature maps, a fully-connected NN does a prediction, which can be a classification or regression.

that the points in the same region are closely related and that the identified patterns can be found translated in the space. Therefore, it is reasonable to use CNN in problems where the input order and location matter. A very well-known example is the successful application of CNN to image processing problems, as evidenced by the numerous times it has been used to gain *ImageNet* challenge [207]. CNN can also be handy to analyze time-series as a 1D array or even by transforming the time-series into images, usually by applying time-frequency transformations.

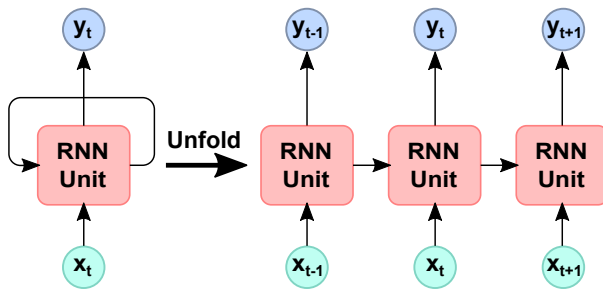
**Convolutional Layer** is a key element of CNN to explore the spatial patterns. It consists of applying a kernel filter to all local regions by sliding it throughout the image, what is called *convolving*. The kernel filter is just a matrix of weights associated with a feature. The sum of the element-wise multiplication of the filter matrix with the local region values is the resultant value that identifies how strongly the feature is detected in this region. Therefore, as it slides through the image, a *feature map* is created, as illustrated in Figure 3. Each convolutional layer can contain multiple filters, each of which results in a feature map which is stacked to each other along the depth dimension to form the layer output. The filtered output will also go through some nonlinear activation functions, being ReLU the most common one. The convolving procedure considers the **local connections** into each region defined by the size of the filters. Besides that, as the filter has *shared weights* for the entire image at each depth slice, the same feature will be detected

at distinct locations, which contributes to the **translational invariance** properties in the network structure. In this way, the convolutional layer makes use of these two important characteristics of spatial signals.

In the sequence, the **Pooling Layer** subsample the feature map shrinking the image stack, usually by calculating the maximum value for non-overlapping patches of the feature map (*max-pooling*) (Figure 3). The output has a pattern like the input, but with smaller dimensions. This stage collaborates to reduce computational load and number of parameters, also avoiding overfitting. However, the pooling layer also plays a significant role in the identification of spatial patterns, since it will semantically merge similar features and add invariance to small scaling, shifts and distortions [195]. In that way, the first layers in a **Deep CNN** are responsible for detecting simple features, such as edges and corners in images and as the CNN depth increase, higher-order features and complex patterns can be detected. In the end, **Fully Connected Layers** perform the final classification or regression task. The backpropagation algorithm is normally used during optimization to adjust the weights from the kernel filters (automatically defining the prominent features to be detected) and from the fully connected layers.

Comprehensive explanations on CNN methods are available in the main textbooks of Deep Learning [17, 183, 189, 192, 194] and visual and iterative guide is provided by [208]. Recent progress was covered in [209]. Transfer Learning is also an important topic in CNN

a-)Rolled and Unrolled RNN



b-)Long Short-Term Memory Network

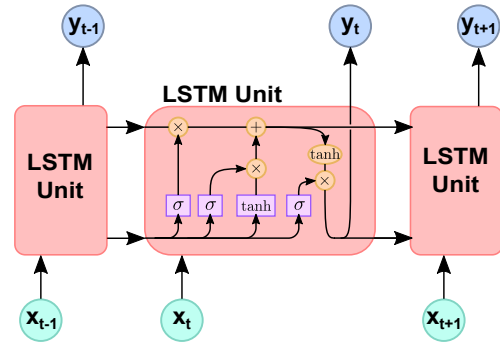


Figure 4: Recurrent Neural Network rolled and unrolled, showing how the network uses the state from the previous time step as information to the current step and so forth (a). Long Short-Term Memory Network with forget, input and output gates forming the memory unit cell that avoids vanishing gradient problem (b).

[17, 204] since features learned from other datasets can be useful to new problems, therefore acclaimed CNN models trained with big datasets, such as LeNet, ZF Net, AlexNet, VGGNet, GoogLeNet and ResNet, can be used as a start point to initialize new CNNs, reducing the need for data.

### 2.1.2. Recurrent Neural Networks

Recurrent Neural Networks (RNN) is a group of NN designed to handle **dynamical systems**, that is, systems changing over time. While ordinary NN do not use sequential information to do predictions, RNNs can use the time history and context. Because of this, they are extensively used in natural language processing, like chatbots, machine translations, speech recognition and generation, in Computer Vision applications and forecasting problems in the most diverse fields [210], as in economy, climate change, weather and diseases prognosis. Naturally, they are also applied to analyze dynamic systems in mechanics, for example, in the prediction of the system response in function of time [211], or to forecast the remaining useful life of a component [212].

When creating an RNN model, the input is configured as a 2D tensor containing the *time steps* in one dimension and the *input features* in the other. To have *memory* ability, RNNs are constructed in loops over the time steps. For each time step, the correspondent input features are provided alongside the current *state* of the problem to obtain the current output and this output is used as the state of the next time-step, configuring the loop [194]. Therefore, the *state* information will carry historical information through the RNN by assuming that outputs of different time steps are dependent on each other. This looping behavior can also be interpreted as a chain of NN which pass information

to the follower [213], as illustrated in Figure 4. However, these sequential NN share the same weights, which are trained by a process called *backpropagation through time*. Usually, RNN suffers from exploding or vanishing gradients over time, but advanced techniques are employed to overcome the issue [195].

Another frequent problem is that over time steps the level of accumulated historical information increases and the RNN cannot define which is the valuable information to keep and suffers from a long-term dependency problem. **Long short-term memory (LSTM)** network [214] addresses this problem by using a memory unit cell with forget, input and output gates which explicitly defines what are the relevant information to keep and to output, as illustrated in Figure 4, showing more effective results than ordinary RNN [195]. The reader can look for a good introductory explanation about LSTM functioning provided by Olah [213].

There are many configurations of RNN, to deal with different dimensions of inputs and outputs, or bi-directional RNN to address cases where previous outputs in the network might be dependent on "future" results, or other configurations to deal with the long-term dependency problem, as the Gated recurrent units (GRUs). Generative RNN has also been extensively used, in special for text and speech generation [194]. More details on RNN theory and its architectures are available in the textbook by Graves [215], in Chapter 10 by Goodfellow et al. [183], Chapter 13 by Bishop [189] as well as in [194, 195]. Recent advances in RNN are reviewed by Salehinejad et al. [216], as new RNN employing the attention mechanism [217].



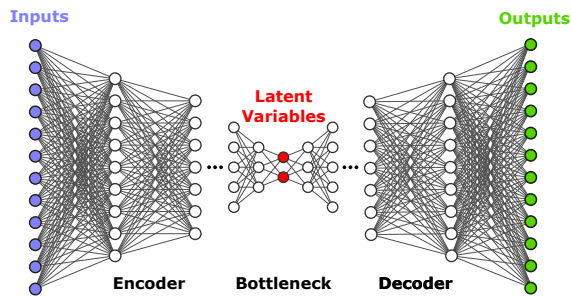


Figure 5: Illustration of an Autoencoder. The encoder stage compresses the information in the latent variables and the decoder stages decompress them. The autoencoder outputs should match the inputs, therefore it is a self-supervised learning.

### 2.1.3. Auto-encoders

**Dimensionality reduction** are techniques which reduce data dimension while preserving the critical information on it. Due to the curse of dimensionality, this is the desired tool which allows reducing processing and storage requirements, especially in problems involving high-dimension data. As an example, dimensionality reduction is often used to create *Reduced order models (ROMs)* to accelerate complex and large numerical models of dynamical and control systems [218, 219]. Dimensionality reduction methods are also extensively used as a **data preprocessing** stage previously to another ML method, since they automatically perform feature extraction or/and selection, improving the predictions of classic classification and regression algorithms.

Several unsupervised ML methods perform dimensionality reduction [220], the principal ones being Auto-encoders, Boltzmann Machines and Principal component analysis. For sake of brevity, the authors are going to cover only Auto-encoders in this article, which have a straightforward concept and are preeminent to nonlinear dimensionality reduction [221].

Autoencoders (AEs) are merely Neural Networks that receive high-dimensional data and encode it to a latent space representation by compressing the information through the NN with decreasing layer size until a bottleneck. The compressed information is then used to reconstruct the original input in a *self-supervised learning*, where the input itself is also used as the target output, as illustrated in Figure 5. Therefore, the data in the latent space must contain the more relevant information of the original data to be able to reconstruct it. Once the AE is trained, the encoder part is used as a dimension reduction tool.

An advanced version of an AE is the Variational autoencoder (VAE) which compresses the data into the mean and variance of a statistical distribution. During

training a random sample of this distribution is selected to reconstruct the original data, which forces robustness and meaningful representations in the latent space and leads to better performance than ordinary AEs [194]. Chollet [194] and Goodfellow et al. [183] presents detailed AE and VAE methodology and emphasize the prominence of using VAE decoders as generative models.

### 2.2. Decision Tree Ensemble Methods

Decision Trees are statistical learning algorithms that apply recursive partitions of the space to perform classification or regression and that are the basic element of more complex and popular models, like Random Forest and Gradient Boosting [222]. There are different Decision Tree algorithms, CART (Classification and Regression Trees) being the most famous. The CART algorithm performs successively binary partitions starting from a root node. The partitions are based on the *impurity criterion* and aim to gather samples from the same category or with approximate value in the final nodes of the tree, also called leaves. Random variables are tested as partition criteria, and the split is performed with the one that minimizes impurity. As finding the optimal of each node would be expensive, the algorithm implementation is greedy, using heuristics methods. The *scikit-learn* Python library provides an easy implementation of Decision Trees, as well as good documentation on the methodology, applied [223].

Decision Trees are interpretable, fast, handle heterogeneous data and outliers, exempt data scaling, implicitly perform feature selection, and are nonparametric, allowing models with nonlinear complex relations [224]. However, Decision Trees are prone to overfitting, leading to high-variance and greedy solutions [17]. Although *pruning* is a regularization technique commonly applied to reduce this overfitting problem[189], it is through ensemble methods that Decision Tree models achieve good generalization.

**Ensemble methods** are a combination of individual predictors to yield a better predictor. Any ML method can be used to create ensemble methods, but ensembles of Decision Trees are popular since Decision Tree randomized structure and low-cost training make them a suitable predictor. Bagging and Boosting are reviewed here, the two most popular ensemble methods. In **Bagging** each predictor collaborates with a vote in the final prediction, and the predictors are constructed independently in a parallelized framework, reducing mainly the variance, as is the case of Random Forest. In **Boosting** the predictors are connected and sequentially organized,

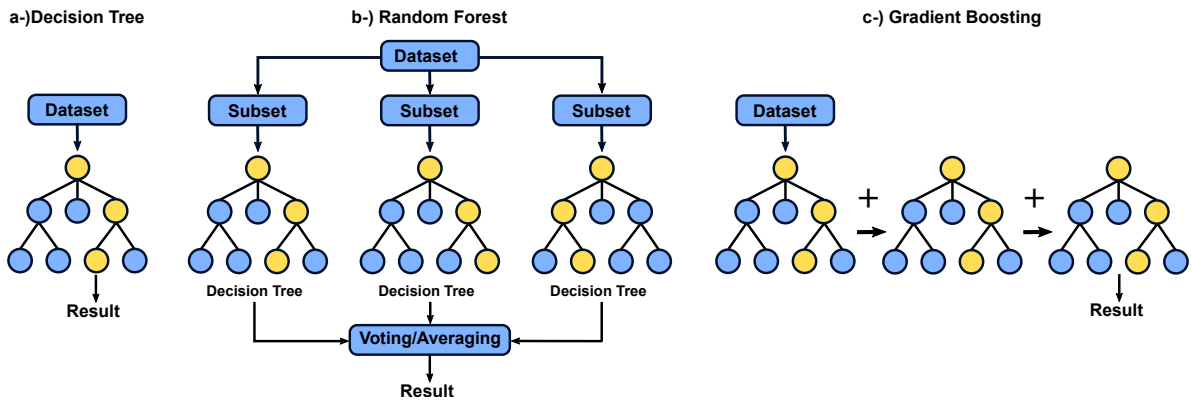


Figure 6: Individual decision tree partitions the space minimizing the classification/regression impurity (a). Random Forest uses Bagging to ensemble decision trees and get the final prediction by majority or average voting (b). Gradient Boosting ensembles decision trees in sequence, so that the individual prediction of one decision tree is improved by the next and so on (c).

each one trying to improve the prediction of the previous, reducing mainly the bias, as is the case of Gradient Boosting. Louppe [224], Chapter 8 by Mehta et al. [17] and Chapter 14 by Bishop [189] present the theory of ensemble methods and its close relation with the bias-variance tradeoff.

### 2.2.1. Random Forest

Random Forest (RF) is an ensemble method based on Decision Trees and *Bagging*. To improve the performance of the individual tree, RF grows several estimators or a forest of trees and makes the final prediction based on the average result of the individual trees, for regression problems, or based on the majority vote, for classification problems, reducing the prediction variability. Each Decision Tree should be different to ensure that the final RF generalizes better than individual Trees. Because of this, the Bagging or bootstrap aggregation procedure is applied, meaning that each Decision Tree is constructed based on random sampling with replacement of the dataset and random selection of subset features. A straightforward implementation of RG is possible with *scikit-learn* [223].

In comparison with NNs [225], RFs are cheaper during training and prediction, demand fewer data, and are more robust to missing data and hyperparameters choice, besides maintaining the advantages of a single Decision Tree. Another convenience of RF is that they intrinsically perform a sensitivity analysis, called out-of-bag (OOB) based sensitivity-index and are more interpretable than NNs [151, 226]. The sensitivity analysis loses accuracy if the prediction is inaccurate or if the problem has highly correlated features. Both Random Forest and NN can model nonlinear complex relations,

but NN usually does it better if big data is available. Besides that, RF has the big drawback of not extrapolating.

### 2.2.2. Gradient Boosting Trees

Gradient Boosting (GB) trees are also an ensemble of decision trees based on the Boosting technique. *Boosting* algorithms rely on the idea that weak predictors can be sequentially added to each other to result in a stronger predictor [227]. Thus, a sequence of decision trees results in a stronger predictor, which also counts with a differentiable loss function, allowing a gradient-based optimization of the Decision Tree parameters. This concept is the base for creating GB methods.

In a regression problem, the first tree predicts the mean of the data, while the next tree tries to predict the residuals of the prediction of the first tree while minimizing the loss function, and so forth. Therefore, the sum of the prediction of all the trees in the sequence will minimize the error and the problem complexity, in a process called *Additive Training*. There are two main powerful methodologies to implement GB trees: **XGBoost** (eXtreme Gradient Boosting) [228] and **AdaBoost** (Adaptive Boosting) [229], which also count with handy Python libraries and tutorials [223, 230]

GB methods usually outperform RF [194], with the downside that the GB algorithm cannot be fully parallelized and does not extrapolate well. Currently, GB methods such as XGBoost and AdaBoost are, alongside NN, the most used ML method in the competitions on the Kaggle website, which is used as an indicator of ML methods relevance [194].

### 2.3. Gaussian Process

Gaussian Processes (GPs) in ML are algorithms that use **Bayesian inference** to update the GP modeling of

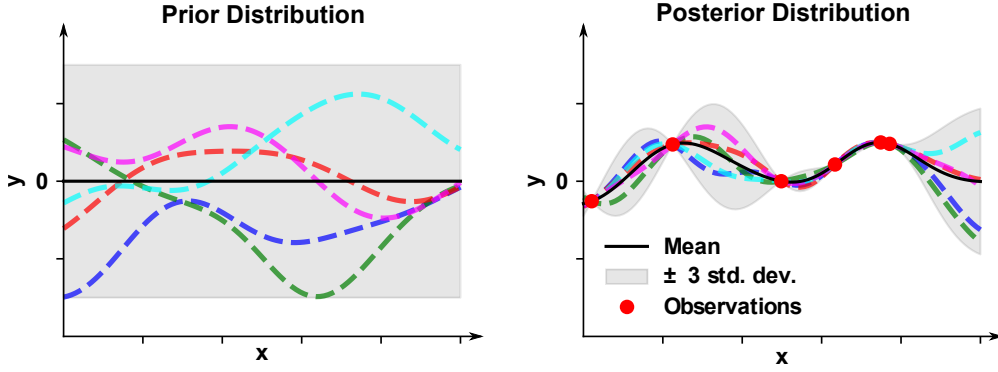


Figure 7: In Gaussian Process Regression, the prior distribution (left) is defined by kernels functions and the posterior distribution (right) is updated with information from the observation points using Bayesian inference.

the prior and posterior distribution until it matches the available data [231]. GP regressors are also commonly referred to as *Krigging*. The GP is a stochastic process that assumes a joint Gaussian distribution over all variables and, thus, a distribution over functions:

$$\begin{aligned}
 f_{GP}(t) &\sim GP(m(t), k(t, t')) \\
 m(t) &= E[f_{GP}(t)] \\
 k(t, t') &= E[(f_{GP}(t) - m(t))(f_{GP}(t') - m(t'))]
 \end{aligned}
 \tag{1}$$

where  $m$  is the *mean function* and  $k$  is the *covariance function* or kernel, which models the correlation between the variables  $(t, t')$ . Now, considering that  $X$  and  $Y$  are in input and output vectors, respectively, of the training points and assuming a zero mean GP with a matrix of covariance given by  $K(X, X)$ , the posterior distribution for any set of function inputs  $X^*$  is inferred as:

$$\begin{aligned}
 f_{GP}(x^*) | [f_{GP}(X) = Y] &\sim \mathcal{N}(m_c, k_c) \\
 m_c &= K(x^*, X) \cdot K(X, X)^{-1} \cdot Y \\
 k_c &= K(x^*, x^*) + K(x^*, X) \cdot K(X, X)^{-1} \cdot K(X, x^*)
 \end{aligned}
 \tag{2}$$

Figure 7 illustrates the process of updating the prior with input data and evaluation of the posterior distribution. Usually, a maximum likelihood estimator approach is applied to optimize the kernel hyperparameters. In other words, the searching process looks for the best fit of the output data to the input data evaluated by the marginal likelihood of data. A detailed explanation, also including the formulation for GP with non-zero mean and inference considering a Gaussian white noise in the observation's outputs, can be found in the works by Beckers [231] and Bachstein [206]. The book by Rasmussen [232] is also a classic reference for Gaussian Processes. Implementation is also widely supported in

many libraries in python, like *scikit-learn* [223], *GPpy*, *GPflow*, *SMT* [233] and *GPpyTorch*.

GP belongs to the class of **probabilistic surrogates** that predict a distribution of the outputs, which makes it possible to access the prediction variance, often used as a level of trust in the prediction. The variance information is used in *informed decision-making* and in *adaptive sampling techniques* that evaluate where to sample new training points. Abdar et al. [234] and Bachstein [206] do an extensive review of several probabilistic ML methods besides GP, such as Bayesian Neural Networks, Monte Carlo Drop-out, Deep Ensembles, Dropout Ensembles, Quantile Regression, etc.

Although several recent techniques try to overcome the problem, GP typically does not scale well with big data and therefore is commonly applied when the input dimension is small and when a limited amount of data is available, such as to construct surrogate models, where GP probabilistic properties are useful. **Bayesian Optimization** is a field that makes extensive use of the GP as a surrogate model in the optimization of expensive functions [235, 236].

#### 2.4. K-means

Clustering is an unsupervised learning method that aims to group data into similar groups, or *clusters*, based on some similitude or distance measurement [17]. Clustering is extensively used to explore data and discover its structure when little or no previous knowledge is available, being especially relevant in *Data Mining*. Many clustering methods have already been developed based on different principles and are appropriate for a wide variety of cases, like K-means, support-vector clustering, hierarchical clustering, mixture models, density-based. Xu and Wunsch [237] present a survey on several clustering methods, while the implemen-

tation of several of these methods is provided by Scikit-learn libraries [223] alongside documentation with tutorials and a summary of the appropriate use cases [238].

The K-means algorithm is one of the simplest and most used clustering algorithms. K-means is a centroid-based method that divides data into a pre-defined number of  $k$  disjoint clusters, minimizing the Euclidean distance between each cluster sample and the cluster centroid, also known as cluster moment of inertia. According to Mehta et al. [17], this can be interpreted as the minimization of the variance within each cluster.

The algorithm consists of random initialization of the centroids after which the algorithm iterates through a loop that first assigns each sample to its nearest centroid and then updates the centroids as the mean value of the samples assigned to it until the change in the position of the centroid is smaller than a defined threshold. K-means is very efficient and scales well for big data, however, it performs poorly for irregular and elongated clusters, it is sensitive to initialization and outliers, and it requires a pre-defined number of clusters.

### 2.5. Reinforcement Learning

Reinforcement Learning (RL) is a class of ML algorithms in which an agent interacts with an environment and learns from the success and errors of these experiences. The agent *actions*  $A$  transform the environment *state*  $S$ , which generates a *reward* (Figure 8). The goal of RL algorithms is to find the optimal sequence of actions that maximize the Quality function (or value function), which is a function that models the expected long-term cumulative reward [239]. The classical book by Sutton and Barto [239] explains the methodology behind RL, and the work by Li [240] presents an overview of Deep RL aspects, applications, and relevant references. The **Q-learning** algorithm developed by Watkins [241] is a popular RL algorithm used here to illustrate the RL workflow.

**Q-learning** is an off-policy algorithm which uses *temporal-difference* learning (*bootstrap* approach) to learn the optimal sequence of actions that maximize the Q-function based on the Bellman equation. Therefore, although a partly random policy is used to select the actions to update the optimal Q-function, the learned optimal action-value function is independent of a policy that determines the agent actions [239, 242]. A Q-table commonly represents the Q-function with discrete variables, whereas the use of Deep-NN to approximate the Q-function shows notorious results in Deep RL with discrete and continuous action spaces [243–245].

Recent outcomes with RL have drawn attention to how these algorithms might be a key part of the fu-

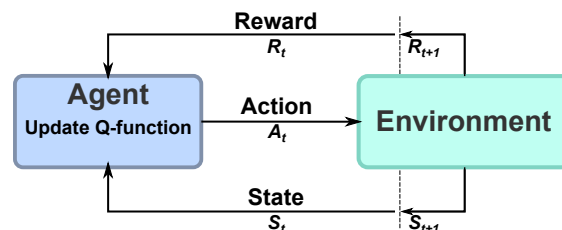


Figure 8: Reinforcement Learning framework: the **agent** performs an action  $A_t$  in an interactive **environment**, resulting in a change from state  $S_t$  to state  $S_{t+1}$  and in a reward  $R_{t+1}$ . The information is used to update the value function (or Q-function in the case of Q-learning), which learns the relation between an action performed in a given state with the reward in the long term. In that way, the agent learns the actions that optimize the system rewards.

ture of artificial intelligence, with notable autonomous machines playing games, controlling robots, recognizing, and classifying images, generating human language content, proposing financial investments, suggesting purchases, among many other applications [240]. In mechanical engineering fields, as the vibroacoustic field, the use of RL to develop adaptive controlling systems is a developing research area, as further discussed in Section 4.2.

## 3. Structural Health Monitoring by Machine Learning

SHM deals with the detection, diagnosis, and prognosis of recipient failures, as well as the prediction of the Remaining Useful Life (RUL) of engineering structures based on its measurements. The benefits of SHM are manifold and well known to structural reliability and integrity management, including avoiding catastrophic failures, optimizing service time with scheduled maintenance, planning missions to minimize wear, tracking the cause of failures, and reusing healthy components.

Data-driven statistical methods are suitable for SHM as they incorporate information of the true operating conditions and the uncertainties involved, while physical-based models usually fail to model the real conditions of the system, often unknown and commonly involve costly simulations, making them unfeasible for some online SHM applications [246]. The growing amount of data available after the Big Data revolution and the fast advance of ML as a powerful data analysis method paved the way for ML-based SHM to become a major area of research.

Currently, SHM is the most developed application of ML in engineering and many successful works and methodologies exist in the literature, most including

vibroacoustic-related problems. The book by Farrar and Worden [28] presents an in-depth analysis of all aspects of SHM related to ML, including main applications, data collection and processing, and ML algorithms. More recent surveys on SHM driven by ML were carried out by Khan and Yairi [29], Azimi et al. [30], Toh and Park [31], Lin et al. [32], Bao and Li [33], Malekloo et al. [34], Lei et al. [247] and Zhao et al. [35], most of them focusing on Deep-learning models. The main applications are in Rotating machinery [36, 37], Civil Engineering [38–41], Earthquake engineering [23] and Aerospace Structures [42–45].

The ML methodologies applied in SHM are closely related to the characteristics of the data. Inspection images, vibration signals, or acoustic signals are the three most commonly used data sources. In the review by Azimi et al. [30], 44 references that use images datasets in SHM are summarized, most of them using CNN to take advantage of its image-processing capabilities. However, in vibroacoustic applications, it is usually more convenient to use time-domain signals, which can be continuously monitored, as vibration and noise signals. Avci et al. [39] presents a review of vibration-based health monitoring in civil structures.

It is often necessary to transform vibration and noise signals to reveal their relevant characteristics, as they might be represented best in the frequency or wavelet domain. Because of this, **Data Transformation** is an important pre-processing step in SHM and much literature dedicates to investigating it. Analyses in the *Frequency domain* are usually applied for stationary signals and can be obtained with Fast Fourier Transform (FFT) or bispectrum analysis [46–48]. Time-frequency or *Wavelet domain* analyses are convenient for non-stationary signals, and their methods and aspects are reviewed by Taha et al. [49]. Many new references can also be listed employing different Wavelet transform methods, as Discrete Wavelet Transform (DWT) [50], Wavelet Packet Transform (WPT) for more noise reduction and adaptive resolution [51–54], Morlet Wavelet [55], Short Term Fourier Transform [56, 57], Hilbert-Huang transform [57] and Empirical model decomposition [58].

Vibration and noise signals are often transformed into multiple arrays or images to train a CNN, taking advantage of the CNN ability to extract features from big data. However, the transformation method from signal to image also plays a key role in these cases. For data in the time-domain, the methods include reshaping the time-series as matrices [47, 59], Omnidirectional regeneration [60], Gramian Angular Displacement Field and Markov Transition Field [47]. Many works used the

time-series signals with 1D CNN [66–72]. In frequency domain, the images are created using Dempster-Shafer theory [61], 3D image method [46], or combining multiple sensor data [62, 63]. Wavelet analysis can also create spectrograms to analyses in time-frequency domain [47, 64, 65]. Reshaping statistical features into a matrix was also used to create images [73].

As data in SHM applications are usually high-dimensional, it is also crucial to perform **Feature Extraction and Selection** to reveal and filter the most pertinent information. The classical approach has handcrafted feature extraction followed by order reduction methods and, lastly, the ML model. However, Deep Learning (DL) has been increasingly used in SHM as its depth enables the extraction of relevant features automatically, ending the need for handcrafted methods Khan and Yairi [29]. In this way, even raw data can be input in DL methods due to its ability to learn high complex and nonlinear patterns. On the other hand, DL usually demands more data. The different procedures of feature selection and extraction in both Classical ML and Deep Learning are illustrated in Figure 9.

Classical ML methods often use handcrafted features in cases with little data availability [47, 68, 74, 75, 83]. The extracted features might be in time-domain (e.g., Root Mean Square, Skewness and Kurtosis), in the frequency domain (e.g., Power Bandwidth, Harmonics, and spectral skewness) [47, 248–251] and in the time-frequency domain, where Wavelet Packet Energy-Entropy is commonly used to extract features [52, 252–254]. Other feature extraction methods used in vibroacoustic problems are Multi-Domain Statistical Feature [68, 251, 255–257], Compressed sensing techniques [52, 65] and Histogram of Oriented Gradients for vibration images [60, 258, 259].

After the feature extraction, feature selection is performed to dimension reduction, currently done by an Unsupervised ML method. Varanis and Pederiva [252] compares some feature selection methods and concludes that Linear Discriminant Analysis (LDA) is suitable for non-stationary cases, while Principal component analysis (PCA) is convenient for stationary signals and independent component analysis (ICA) for problems with combined faults. Malekloo et al. [34] also reviews several supervised and unsupervised feature selection methods. The classical ML approach in SHM is addressed by Worden and Manson [260] for damage detection, localization, and assessment problems.

The use of DL algorithms in SHM occurs mainly in two distinct configurations: an Unsupervised DL algorithm to perform dimension reduction, usually Autoencoders (AE) or Boltzmann Machines (BM), fol-

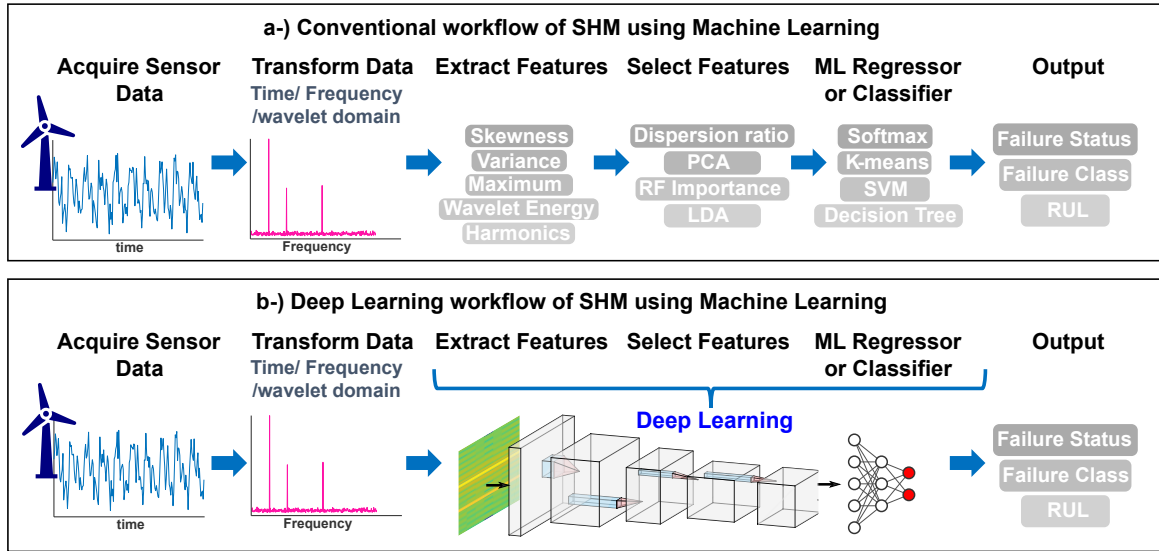


Figure 9: Structural Health Monitoring workflow: in the classic approach, feature extraction and selection are handcrafted and followed by an ML method (a); if Deep-Learning is used, feature extraction and selection are performed automatically by the ML method (b).

lowed by a simple ML classifier or regressor, which can be either a supervised algorithm (as a decision tree) [54, 74, 80, 93, 261] or an unsupervised algorithm (as k-means) [79, 94]; or a DL algorithm which performs all stages, that is, feature extraction and selection and the final prediction, as the case of CNN or LSTM [47, 76, 77].

Malekloo et al. [34] provides a complete review of all processes that should be considered in an ML-based SHM analysis, from the excitation source, going to data acquisition, data normalization, data cleaning, data compression, feature extraction and selection, data fusion, and the prediction. Hereafter, some ML-based SHM works using vibroacoustic signals are presented to exemplify applications in damage detection and diagnosis - including damage location, extent, and type - and in the prognosis of RUL and mission planning.

### 3.1. Failure Detection and Diagnosis

The first level of complexity of SHM problems is failure detection, that is, identifying if a signal is healthy or not. Unsupervised learning algorithms are suitable for these applications, once the failure can be detected just by identifying anomalies and outliers in the signal, without the need to have labeled data, which is currently the case in real applications. Rizzo et al. [78] show an example of unsupervised learning crack detection using hand-crafted extracted features by discrete wavelet transform and outlier analysis. The article also exemplifies how hand-crafted features can lead to good accu-

racy, but also how laborious it can be to select the best features to be used. Automatic feature extraction by unsupervised deep learning approach is implemented by Reddy et al. [79], that used a deep auto-encoder to extract features and reconstruct the signal from unlabeled datasets with raw and heterogeneous data (coming from different sensors modalities). A threshold in the error of the reconstruction signal was used to identify unhealthy signals.

The next complexity level in SHM is fault diagnosis, which includes the prediction of damage location, extent, and class [34]. For example, in the work of Reddy et al. [79], after detecting the fault, a clustering unsupervised method is applied to classify the fault type. However, supervised methods are more commonly used when the diagnosis of fault location, class and extent is required. Sun et al. [80] used a Sparse Autoencoder to extract features from the vibration signals of an induction motor, followed by a Dropout NN to classify the fault.

Gecgel et al. [47] compared several approaches to identify and classify gear tooth crack based on simulated-based vibration signals. Various levels of tooth profile error were also considered, and the noise was artificially added to the signals to augment the robustness. Gecgel et al. [47] performed the classical approach of feature extraction followed by classical ML methods and compared it with Deep-Learning approaches without feature extraction, as CNN and LSTM. The DL methods overcome the classical ap-

proach for all ML algorithms, while among the DL methods, CNN beat LSTM. However, Gecgel et al. [47] shows that the CNN accuracy level is highly dependent on the preprocessing method used to encode the vibration signals into images.

As indicated by these examples, vibration signals are the most used inputs in ML-based SHM problems, and their many uses are reviewed by Toh and Park [31]. ML-based methods can also use acoustic emission signals to detect and diagnose failures. The reader can consult the recent work by Suwansin and Phasukkit [81] or the review by Muir et al. [82] to have more details and references, being that the latter focus on diagnosis of composites structures.

Acoustic measurements can also be used as non-intrusive sensors to failure diagnosis, as investigated by Janssen and Arteaga [46]. The signals from a microphone array placed close to a vibrating plate were used to localize a failure in the plate, which was experimentally represented by a disturbance mass attached to the plate. Janssen and Arteaga [46] investigate processing and data augmentation methods, as well as the number of microphones needed for an acceptable accuracy in the location prediction. The psycho-acoustic features extracted from acoustic signals also demonstrated to be meaningful inputs for failure detection in gears [83], which can also be indicative of NVH performance. Li et al. [54] implemented a method to merge acoustic emission and vibration signals by extracting the signal features through separately Deep Boltzmann Machines and merging them with a Random Forest, showing improved accuracy in the classification of gearbox failures in comparison to other approaches.

Hybrid models integrating physics-based and ML-based models are also found in the literature for fault detection and diagnosis. Abbiati et al. [84] implemented a Hybrid Model to detect Euler buckling failure in a beam using Kriging meta-models and active learning to assess structural reliability. The Hybrid model proposed by Ritto and Rochinha [85] configures a Digital Twin due to its bi-directional connection, which allows the model to be calibrated with data from the physical twin (real asset) and the digital twin predictions can be used to update the physical twin operation parameters and control strategy. The methodology is demonstrated in a bar structure, which is trained offline using displacement synthetic data to detect damage and identify its severity and location. The study also analyses the robustness of the DT to different damage intensities, noise and uncertainties levels, the number of sensors and the excitation frequency and location.

### 3.2. Prognosis and Mission Planning

More than detection and diagnostics, the potential of the SHM relies on prognosis. Predictive analyses allow us to estimate the RUL and therefore, it can be used to optimize the maintenance schedule, minimize downtime, and lead to money savings more safety operation conditions. Si et al. [86] and Jardine et al. [87] provide a review of the Data-Driven methods to evaluate RUL and apply Condition-Based Monitoring.

When degradation historical data is available, the DT can predict the RUL just based on online-sensor data and in the component operation conditions. As an example, an LSTM-based model with an attention mechanism was implemented by Muneer et al. [88] to predict RUL curves with uncertainties using a turbo-fan benchmark dataset, outperforming CNN, RNN, and GRU models. Zhao and Yuan [89] implemented a CNN which detects and classifies faults in the outer race, inner race, and the cage of a bearing and, once the fault is detected, the DT predicts its RUL in real-time with an online adaptive delay correction method to obtain an improved accuracy result. Other examples include a semi-supervised approach using VAE and RNN [90], an LSTM with dimension reduction methods for multi-sensor data of machining tools [91], a hybrid model with NN and GPR to predict fatigue failure time with adaptive confidence interval [92], or even just an exponential fit [262].

Similarity models can also be used when only run-to-failure data from other similar components are available. In this case, the RUL is estimated based on a curve with a similar profile, as implemented in MathWorks [263] and Liao et al. [93]. Finally, if no degradation or failure data is available, the failure can be estimated with relation to an established threshold in some of the monitored data, just by predicting its future state.

Booyse et al. [94] shows the possibility of using a dataset only with healthy vibration data to detect damage, indicate damage severity, and identify failure mode. To illustrate the methods, analyses were performed in simulated datasets of a gearbox with tooth damage and of an Aero-Propulsion System with compressor or fan failures, as well as an experimental dataset with bearing failures. An order tracking preprocessing was applied to normalize the data with respect to rotational speed, as well as time-synchronous average over the recording period. Generative Adversarial Network (GAN) and VAE were used as the ML unsupervised algorithms, being that GAN presented the best performance.

Recently, DT models have been proposed in SHM to analyze "what if?" scenarios, perform risk assessments

and optimize the asset operation to maximize its life. Karve et al. [95] implements a DT which performs failure diagnosis and prognosis, as well as optimize the operational parameters of the system, resulting in damage-tolerant planning which minimizes fatigue crack growth while still ensuring that the mission aim is achieved, all considering both aleatory and epistemic errors. Different methods of fusion Physics-based and data-based models are applied in each stage of the DT and experimental results are obtained to validate the methodology.

Kapteyn et al. [96] also developed a data-driven DT to mission planning based on both component and model libraries applied to a fixed-wing unmanned aerial vehicle. All assets share the same physics models, and the analyses are efficiently scaled to the entire system because of the component-based approach, which uses the Static-Condensation Reduced-Basis-Element method. In addition, a library of physical models is used, so that the more appropriate model is applied according to the scenario detected. The physics models use model order reduction to speed up the computation and enable fast predictions. Optimal trees are applied as interpretable classifiers to select the best physical model and update the DT based on the sensor data. Therefore, informed by the level of damage estimated by the DT, the system decides what maneuvers to do, avoiding structural failure.

Stender et al. [97] approaches the acoustic brake squeal problem in two steps: brake NVH assessment and brake squeal prediction. For the NVH assessment, a Short-time Fourier transform creates 2D data representations, which pass through random modification to augment the data and avoid overfitting before training a CNN. The trained CNN identifies the class of brake noise and indicates when and in which frequency it occurs with accuracy. In the second task, the problem parameters over time are the inputs of an LSTM algorithm that predicts when the squeal will occur. However, this methodology showed poor performance when the brake configuration under analysis differentiates from the brake used during LSTM training.

#### 4. Active Control powered by Machine Learning

Active Control is the area of study which aims to model dynamic systems and design control mechanisms to guide the system behavior to the desired state. Active Vibration Control (AVC) of flexible structures plays a significant role in the safety and ergonomics of vehicles, aircraft, machines, and buildings as in manufacturing accuracy [264–267]. Active Noise Control (ANC) or noise-canceling is based on destructive interference

and is a subject of longstanding research [268, 269]. The growing importance of user comfort and product development focused on ergonomic and NVH performance [270] increase the efforts to control vibration and noise. The need for AVC and ANC is greater in the low-frequency ranges, where the application of passive control is limited [271].

Active Control and ML are deeply correlated. Both are data-based science, which development relates to the popularization of sensors and IoT devices, improvements in signal processing, and growth of computational memory and power. In addition, ML methods can be applied in several elements of a control system, as analyzed in the book by Brunton and Kutz [272].

The Least Mean Square (LMS) filter, used in control to estimate the state and controller parameters, is a basic linear ML algorithm. More advanced ML algorithms, in special NN, are also widely used, in special to model and control **nonlinear** systems in which linear control theory might fail. When possible, linear control methods should still be prioritized, because of their smaller response time and the well-developed control algorithms suited to them. Back to the 1990s, many NN applications in active control have already been identified with three usual configurations[273–275]: NN-based Model Predictive Control (MPC), in which an NN black-box models the forward dynamics of the system [276]; as an NN-based model-free controller [277]; and a model reference control, where NN models the plant and optimizes the controller parameters [278]. The first and third, use NN in the system modeling stage, as analyzed in 4.1, while the second and third configurations use NN to learn the optimal control design, as discussed in 4.2.

Brunton’s series of videos, named ”Data-Driven Control with Machine Learning” [279], covers overall aspects of ML applied in Active control. In a recent article, Brunton et al. [22] also reviewed the ML methods applied to fluid dynamics control. Kim et al. [280] reviews the ML methods applied in the control of soft robots, with a focus on soft sensors and actuators. This paper explores the methods which have been applied to the vibroacoustic domain, in special ML-based methods supporting dynamic system modeling, using system identification and reduced-order models, intelligent placement of sensors and actuators, and adaptive control algorithm. Within this topic, it is included the control of buildings vibration under seismic activity, also reviewed by Xie et al. [23]. An extensive, but not an exhaustive number of applications of ML in ANC and AVC is illustrated in Figure 10.



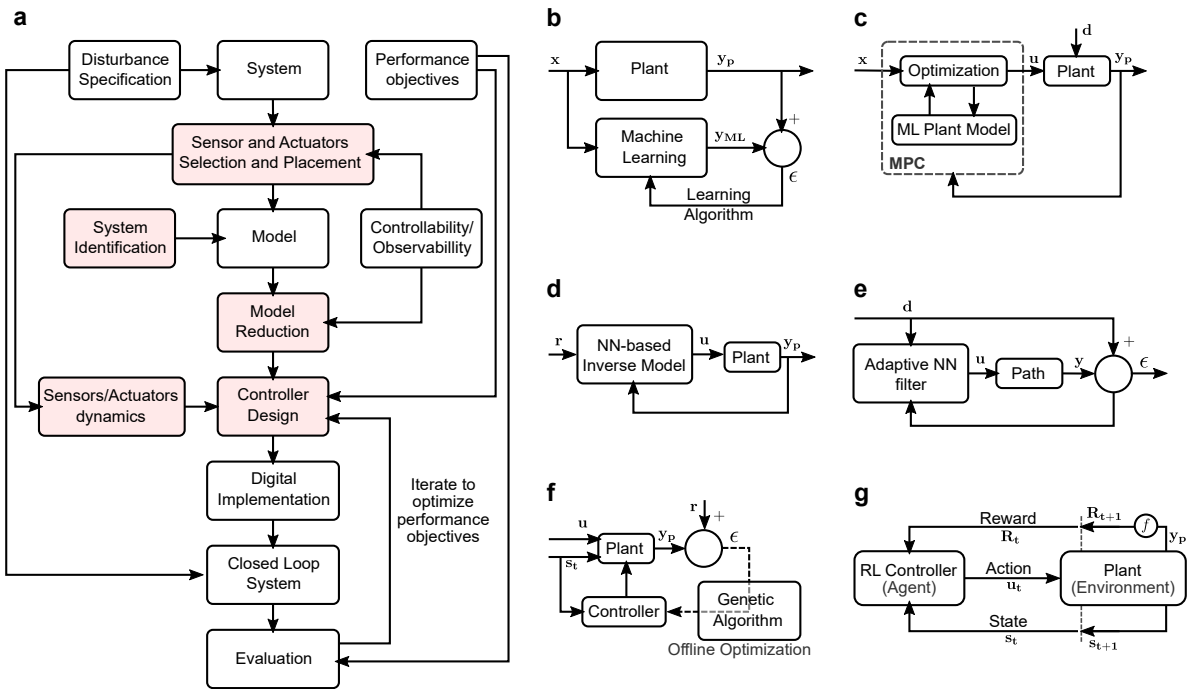


Figure 10: Applications of Active Control of vibrations and noise powered by Machine Learning (ML). (a) Active Control workflow highlighting with red shadow the process that can use ML; (b) Scheme of ML training in **System Identification** problems; (c) Model Predictive Control (MPC) based on ML-model; (d) Controller design using NN-based Inverse Model; (e) Active control using Adaptive NN filter to define control parameters; (f) ML Control using heuristic methods to optimize control structure and parameters; (g).

#### 4.1. Dynamic System Modeling with Machine Learning

A big part of Active control theory relies on *model-based* control techniques in which control needs to have a mathematical model of the physical system, as in model predictive control and linear optimal control [272]. However, in practical situations, there are two main obstacles:

- The physical model of the system is unknown, or the model parameters which fit the system equation are unknown. In this case, it is necessary to apply **system identification** techniques.
- The physical model is known, but its complexity is unfeasible for real-time control applications. Here, it is necessary to apply **model reduction** techniques.

##### 4.1.1. System Identification

System identification (SI) are techniques that use the measured data of a system to model the relationship between the input and output of the system. This description coincides with the definition of ML models of inferring model from data [272, 281]. Therefore, according to Duriez et al. [282], classical system identification

methods - as eigensystem realization algorithm, Kalman filters, and linear parameter varying - can be considered an early form of ML. From this, it can be inferred that modern ML methods should be used for more complex and nonlinear SI.

Chiuso and Pillonetto [283] and Pillonetto et al. [284] did a review of SI methods based on ML, especially the so-called kernel-based methods, highlighting their continuous structure selection capabilities about classical SI methods. A comparison of the online learning performance of adaptive filters in an ANC application, showed the superiority of kernel-based models, as Kernel-LMS and Kernel Affine Projection Algorithms, over classical LMS and NN algorithms [285, 286].

However, NNs have been similarly applied to classical SI methods since they are nonlinear autoregressive exogenous (NARX) models [281]. The sequential dynamic structure of RNNs is also suitable for SI of dynamics systems [287]. citet|jung2020deep analyzes the similarity of Deep Learning and SI concepts and shows the workflow and the results of implementing an LSTM in the identification of a nonlinear state-space model. NN black-boxes are implemented in AVC of vehicle suspension Vidya and Dharmana [98], Xu and Fei [99]

and buildings [100].

Another advantage of NNs over classical SI models is that, besides being able to model nonlinear dynamics, they can be trained to predict the model output some steps ahead. This method is convenient in Model Predictive Control (MPC), which uses a prediction of the system response based on a model of the plant to optimize the control signal over a finite-time horizon in relation to the control cost function in a feedforward configuration. Jamil et al. [100] uses an NN Predictive Control applied to a tall building, combining the good aspects of pole-placement and Neuro-fuzzy control.

Although state-space representations can be obtained from ML black-box models [100], they may also be constructed based on first-principle knowledge of the system dynamics, as with Kalman filters. ML-based state-space models which use both data and first-principles are found in the literature [288–290], but the authors did not find applications in vibroacoustic.

The use of Koopman theory in dynamic mode decomposition (DMD) to describe a nonlinear system on a linear basis has also been leveraged by ML methods, especially in fluid dynamics [22, 291, 292], allowing the use of advanced linear control methods. Recently, Saito and Kuno [101] investigated the application of Data-driven DMD in structural dynamics problems. The sparse identification of nonlinear dynamics (SINDy) method proposed by Brunton et al. [293] to discover governing equation using ML and sparsity techniques have also been used to identify the structural dynamics equations of a geometrically nonlinear system [294], but it was not implemented in AVC.

#### 4.1.2. Reduced Order Models and Sensors/Actuators Placement

**Reduced Order Models** (ROM) are also essential to construct efficient real-time control, since they use lower-rank representations of the system without losing valuable information about its dynamics, balancing accuracy, and efficiency. In that way, ROMs lead to reduced response time and memory requirements, key aspects in active control. Once more, ML methods play a significant role, since there is an intrinsic relation between ML and many ROM.

The first scenario in which ROMs are applied in control is when there is a high dimensional numerical model of the system, as a Finite Element Model, which is computationally expensive for real-time applications. Therefore, ROMs or meta-models are used to speed up the simulation of the system prediction in model-based control. Feedback control might require further reduction in the space-state representation. Besides the ML-

based surrogate models presented in section 5, Component mode synthesis (CMS) are used for linear ROM of structures from FEM models, while ROM based on Proper Orthogonal Decomposition (POD) can be obtained directly from measured data [22, 109].

POD, also known as Principal component analysis (PCA), is also an ML method, once it is equivalent to a symmetric autoencoder with a linear activation function, as demonstrated by Baldi and Hornik [295]. The POD/PAD applies a coordinate transformation from the physical coordinates to an orthonormal basis formed by the system eigenvectors. By selecting only the main modal contributions, or first principal components, the system model is represented on a reduced basis, which is very convenient to model acoustic and structural dynamic problems and has for long been used in space-state vibroacoustic control [102–106].

This modal basis representation also provides useful information on the controllability, observability, and stability of the system, which are key factors to determine the optimal **placement of sensors and actuators** and therefore PCA has also been used for this purpose [103, 107]. The location of sensors and actuators is a crucial aspect in active control design since it influences the control efficiency, cost, and stability [267].

Real-time predictive control applies other tools combining ML methods with ROM. The following examples explore these techniques, which could be used in online control. Liu et al. [108] developed an automatic FEM model updating by using CMS together with Kriging. Simpson et al. [109] used autoencoder to get nonlinear ROM (or Nonlinear normal modes) of a frame structure with hysteresis and used it alongside an LSTM model to predict the system dynamics in near real-time. A Deep Learning-based ROM of structural dynamic systems with inertia and geometric nonlinearities is implemented and benchmarked with simulations of a Doubly clamped beam resonator and a MEMS Micromirror [110]. Using cluster-based ROM, already explored in fluid control [296] and static structural mechanics [218], could have potential use in the vibroacoustic domain.

#### 4.2. ML driven Control Design

Another application of ML methods is in the **control design**, that is, in optimizing the control signal or control laws regarding the cost function that quantifies the control performance. While in the last section ML computes the forward output of the system, the following references use ML to learn effective control laws. ML-based controllers are mainly used to handle systems with nonlinearities, especially using NN methods, as evidenced in the survey on Nonlinear ANC by Lu

et al. [111]. Several configurations use ML to support the control design, such as Model Reference Control, inverse-dynamics control, Machine Learning Control, neuro-fuzzy control, and Reinforcement Control.

In **NN-based Model Reference Control**, two NNs form the control system: one NN plant model predicts the system response and the other NN defines the controller parameters which are optimized to minimize the error between system response and the reference signal. Vidya and Dharmana [98] implemented a Model Reference Control of a vehicle suspension using an NN reference model and an RNN controller, claiming that it leads to better adaptivity and stability. The drawback of NN reference control is that it uses dynamic backpropagation in the optimization, which is computationally expensive [273].

**Adaptive NN controllers** are popular in noise and vibration control with diverse methodologies. One of them is the NN-based inverse dynamics control, which consists of training an NN to model the system dynamics and using the NN inverse operations to determine the controller parameters, as in a regressor-based control. De Abreu et al. [112] implemented a direct inverse NN control of a vibratory system by training an NN as the inverse model of the plant, such that the NN receives the current state and the desired state and outputs the actuator signal. Similarly, Ariza-Zambrano and Serpa [113] applied direct inverse NN control to a beam cantilever, in which the NN was trained both with a full state FEM model and with a reduced model to account for dynamic uncertainties in practical scenarios, showing more stable results than H-infinity control. Nerves and Krishnan [114] uses NN direct controller to control wind-induced vibrations in a building-TMD system, by considering the plant as the output layer of the NN, as in a feedback linearization control. Bani-Hani [115] uses NN to model both a direct forecasting model and an inverse model also applied in the control of wind-induced vibrations.

Model-free NN controllers have also been used in a different configuration of ANC, as a nonlinear alternative to the commonly used adaptive Filtered-X LMS (FxLMS) algorithm. Park [116] tested different configurations of NN as the adaptive controller in a feedback configuration for different ANC datasets. CNN was the one that performed the best, followed by MLP and RNN, all of them with better performance than typical LMS-based controllers. For the case of a feedforward noise control system with a nonlinear primary path, Zhang et al. [117] also obtained better performance with LSTM based controller than with FxLMS.

Zhang and Wang [118] implement Deep-ANC in a

feedforward configuration in which a Convolutional Recurrent Network (CNR) is used to estimate the optimal control signal-to-noise cancellation. The supervised training of the CNR uses the reference signal as input and the ideal anti-noise as the target, both in their spectrogram format. Besides that, the CNR predicts the canceling signal with some frames in advance, to compensate for the ANC delay. The approach improved noise canceling in noise-only and noisy speech scenarios in comparison to typical ANC. Other examples of NN applications are found in the review by Lu et al. [111] on ANC for nonlinear systems.

Heuristic algorithms, such as genetic algorithm (GA) and particle swarm optimization (PSO), can search for an arbitrary optimal control law in **Machine Learning Control (MLC)**. According to Hansen et al. [271], MLC can optimally adapt the weights of any nonlinear filter structure, including an NN. As MLC does not rely on a fixed structure of the controller, nor on a model of the system, it gives more flexibility to the optimization, with the downside that it adapts slowly, preventing its online application to a transient system. Chapter 2 of Duriez's book [282] has a gentle introduction to MLC. [119] was a pioneer to apply MLC in ANC and AVC and was followed by many others in acoustics [120–126] and vibration [127–131] control.

**Neuro-fuzzy control systems**, especially using adaptive network-based fuzzy inference system (ANFIS), have been widely applied in ANC [132–135] and AVC [136, 137], to name a few works. Neuro-fuzzy systems usually use expert knowledge to set initial fuzzy rules in an NN-like structure, then the neuro-fuzzy parameters can be adapted during a training process to fit measured data. The resultant Neuro-fuzzy systems combine advantages of using interpretable explicit rules from fuzzy rules with the learning capabilities of NN.

Finally, noteworthy results are being achieved by exploring **Reinforcement Learning Control (RLC)** [138]. As explained in Section 2.5, in RL the agent (the controller) can interact with the environment (the dynamic system) and its actions will affect the output of the system and, therefore, the Quality function quantifying the long-term performance of the control, which the algorithm optimizes. In this way, RL interactively learns information about the system and the controller behavior altogether, in a similar fashion to human learning. Detailed RLC explanation and references examples are presented in [138, 297].

RLC has gained prominence in applications such as autonomous car control and robot control [298] but has also shown applicability to control acoustic and structural dynamic systems, special in problems with high

uncertainty and stochastic behavior [138]. Latifi et al. [139] shows a successful example in which they applied RLC to manipulate an acoustic field by controlling a centrally-actuated vibrating plate (Chladni plate) and guided a particle towards a target location on the plate surface. Implemented ANC using Q-learning algorithm also had satisfactory results, showing the capabilities of RLC to adapt, as when the secondary path of the noise changes suddenly [140].

Qiu et al. [141] carried out bending and torsional vibration control via an RL algorithm virtually trained with a validated FEM model and transferred to an experimental setup where it shows better performance than PD control. The vibration control of a rotating machine was also performed through RLC using pad actuators [142]. Gulde et al. [143] implemented a method to compensate vibrations in an industrial machine tool using RLC. Eshkevari et al. [144] and Gao et al. [145] also achieved good controllability of flexible buildings structures through RLC. Although RLC shows potential as a real-time decision-making control for complex and uncertain scenarios, it demands considerable training time and expensive computational resources and, therefore, its use may be superfluous to applications already mastered with simpler solutions.

## 5. Vibroacoustic Product Design by Physics-Driven Surrogates

Physics-Driven Surrogates or Meta-models are simpler and cheaper replicas of a high-fidelity simulation constructed based on the information from some input-output points of the true simulation. They have long been used as practical and efficient tools for decision making and risk management in the early stages of product development, once they make it workable to carry out domain exploration, uncertainty propagation analysis, sensitivity analysis, and optimizations, studies in which many evaluations of the same function are necessary.

In the article "Modelling for Digital Twins — Potential Role of Surrogate Models", Bárkányi et al. [299] coins the term Physics-driven surrogate models (PDSM) for surrogate generated from a large amount of data got from a high-fidelity physical model and enumerates several advantages and applications of DT with surrogate models. First, after training, PDSM is much faster than traditional first-principle simulations. In addition, despite being a "black-box" algorithm, it is guided by the physics of the supporting data. Unlike a data-driven model, PDSM is less susceptible to different bias

sources, and its uncertainties can be related to the physical model, so they can be estimated and bounded. As for drawbacks of the PDSM, Bárkányi et al. [299] mentions the lack of interpretability, inability to extrapolate the prediction to unseen scenarios, and the difficulty of assimilating long-term historical data. However, some built-in or implemented methods can increase the interpretability of ML, and Adaptive Sampling can deal with extrapolation when the prediction uncertainty is high. Hybrid methods to embed physical knowledge in the ML can also alleviate these issues.

Despite being a well-known tool, its use in structural dynamics and acoustic applications is still not well established, as in other physical domains. The reason might be that many vibration and acoustic analyzes present a harsh and discontinuous behavior, especially close to the system resonances, which hinders the ML generalization ability [146, 300]. One of the main assumptions of ML is that data is locally smooth [17, 182]. Therefore, the ML surrogates smooth the system response and underestimate sharp peaks and valleys, regions of interest in vibroacoustic analysis.

Tsokaktsidis et al. [147] uses an NN surrogate to predict the acceleration response of a source-receiver structure as a function of the excitation and geometry, showing overall good agreement, but with some inaccuracies in the peak amplitudes. After going through data reduction techniques, data from a beam acceleration response was clustered and used to train NNs aiming to replace FE models Birky et al. [148], but some prediction curves still present high percentage errors.

Approaches as Adaptive Sampling [301], Physics Guided ML methods [302], global surrogates with local refinements, Domain-decomposition methods [300] and Low-bias ML are suitable for dealing with modeling of rough functions. For example, a Physics-guided convolution neural network, with embedded physical constraints, was used to predict building response under earthquakes excitations [149].

The following sections review the workflow for the construction of the surrogates and their applications in acoustics and structural dynamics analyses. Applications in uncertainty propagation and optimization problems stand out since there is a cross-fertilization of research of these domains with surrogate models [299, 300, 303–305], especially because of the benefits of surrogates when performing several evaluations of the same function and because of their intrinsic statistical characteristics. Besides that, ML may also improve similitude techniques to scale models and prototypes during product design, especially for complex structures with incomplete or distorted data [150].

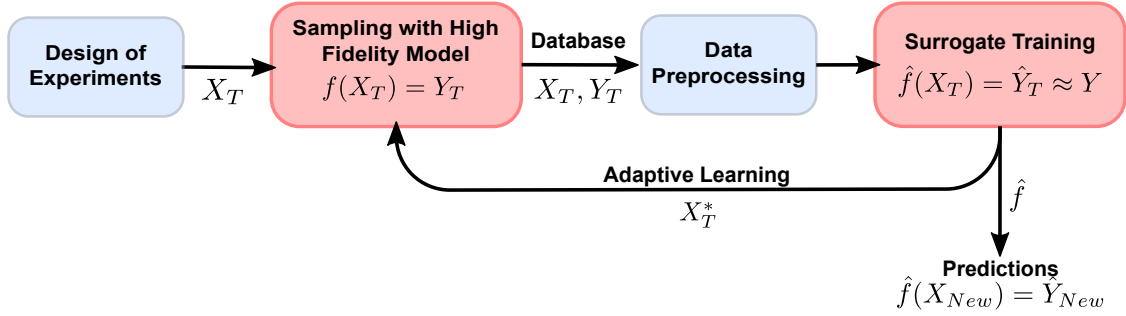


Figure 11: Steps to build a surrogate model: apply DOE to define supporting points location; sample results with the high-fidelity model; preprocess the data; train the surrogate model; predict new outputs using the surrogate model. Adaptive Learning is optional and may be applied to select new supporting points used to update the surrogate model in regions of interest or uncertainty.

### 5.1. Workflow of Surrogate Construction and Related Methods

As the name implies, the Physics-driven surrogate mimics the behavior of a true function  $f(x) = y$ . For this, the surrogate models use statistical methods to map the relationship between a sample of inputs  $\mathcal{X}$  and the corresponding outputs of  $f(\mathcal{X}) = \mathcal{Y}$ , which are called support points. In this way, the surrogate generates a new approximate function  $\tilde{f}(x) \approx f(x)$ , which can generalize the observed behavior of the true function and then predict the outputs for a new set of inputs  $\tilde{y} = \tilde{f}(x)$  with low computational cost and accuracy lost, providing a compromise between computational cost and fidelity. Several statistical methods are used as surrogates, such as Polynomial Chaos [306, 307], Response Surface Model (RSM), Polynomial Function, Radial Basis Function (RBF) [308], Low-rank tensor approximations [304], spectral expansions [300], however, this work focus in ML techniques, capable of model complex and nonlinear relations.

The basic steps to build a surrogate model are schematized in Figure 11. The first step is to generate the support points. As the approximate function is created based on their information, it is of major importance to generate an informative set of support points, according to the Design of Experiments (DOE) theory. Methods as Latin Hypercube Sampling or Quasi-Monte Carlo are used, since they have good space-filling properties, providing information in the entire design domain, including the interaction between parameters, with an affordable amount of sampling points [305].

**Adaptive Sampling** automatically defines regions where to add support points to update the surrogate model, as reviewed by Liu et al. [301]. Adaptive Sampling comprises training a surrogate with a sparse initial database and defining new sampling points based

on so-called Acquisition Functions to query and update the surrogate. In that way, Adaptive Sampling increases the surrogate accuracy near points of interest. This procedure continues until it reaches a stopping criterion, as illustrated in Figure 12, showing the entire workflow of surrogate construction with Adaptive Sampling.

The Acquisition Function, also called Infill Sampling Criteria, accounts for the surrogate mean and variance when choosing enrichment points that should present a compromise between the exploration of new regions (global search), where the surrogate variance is high and the exploitation of promising regions (local search), where the surrogate prediction is of interest, as near the optimal. The surrogate should use a probabilistic ML method once the prediction variance is necessary. Comparisons of Acquisition Functions are presented by Chaiotha and Krityakierne [309] in constrained optimization and by Emmerich et al. [310] in multi-objective optimization.

Figure 12 illustrates a probabilistic surrogate model prediction with the respective Acquisition Function, which determines new observation points. Adaptive Sampling improves the accuracy of the surrogate, especially in regions of interest, minimizing the number of points that must be evaluated with the true function to achieve the required performance and are widely used in optimization problems, as also illustrated in Figure 12 and further discussed in Section 5.3.

Aspects such as dimensionality, data format, presence of outliers, non-linearity aspects of the model and the need for variance information should be considered when choosing the ML method of the surrogate. Although the surrogate can be a classifier, the target is more often composed of continuous values which predict the physical system response. In front of this, most of the supervised regression methods are appropriate to construct surrogates, like Support Vector Ma-

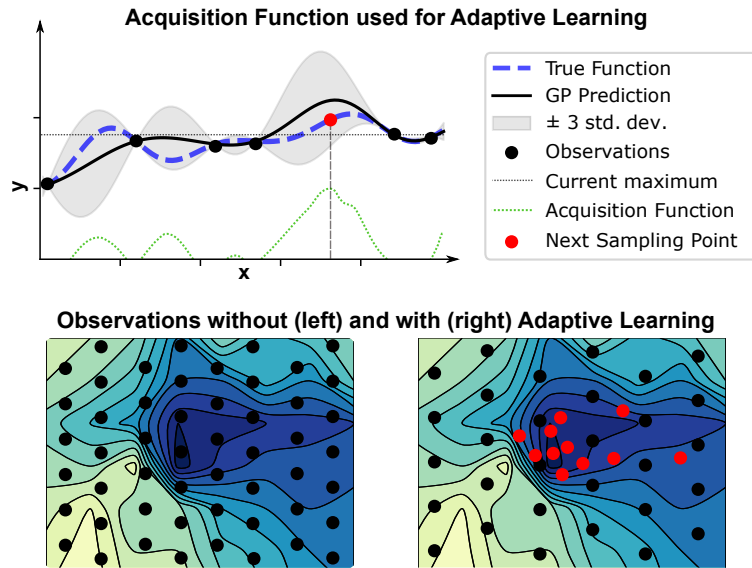


Figure 12: Illustration of the Adaptive Sampling process. The upper image show a Gaussian Process (GP) surrogate model and its Acquisition function used to select new sampling points; the lower images illustrate the initial observation points (black) and the new samples points (red) for the cases with (right) and without (left) Adaptive Sampling.

chines [178], Gaussian Process Regressor or Kriging [311, 312], Neural Networks [303], Random Forest, Gradient Boosting, etc. As surrogate models rarely involve high dimensions inputs, since the ML inputs are the parameters of the physical problem, the choice of Deep Learning methods is not always as advantageous as in SHM applications. However, the limited number of samples and input dimensions make Kriging-based surrogates viable to use and, because of their probabilistic properties, they become a common choice.

Marelli et al. [300] mentions different classes of surrogate models which perform better in different scenarios. Localized surrogates generate predictions that rely on the proximity of the support points and are good at interpolating. Global surrogates have better extrapolation capabilities but usually have lower local accuracy. Global approximations with local refinements or domain-decomposition-based methods are suitable for functions with highly localized behavior in some specific regions of the input space.

## 5.2. Uncertainty Quantification with Surrogates

Uncertainties are an inherent part of every phenomenon and computational analysis. Their study improves comprehension of the phenomenon and enables an adequate level of reliability. The main steps of an Uncertainty Quantification (UQ) analysis are uncertainty propagation and sensitivity analysis, being that both demand several evaluations of the system model and are

used with surrogates (Figure 13).

**Sensitivity analysis** enables us to understand the variability of the outputs by the uncertainties in the inputs and their combinations. The influence of the inputs can be studied separately by performing a local sensitivity analysis (LSA), or simultaneously with a global sensitivity analysis (GSA), which allows capturing effects of the interaction between inputs [151]. Sensitivity analysis improves understanding of system behavior and the interpretability of surrogate models and allows the choice of the most influential inputs for the system response to perform further optimization on a lower-dimensional problem, without major loss of accuracy. Sensitivity techniques can be based on multiple solutions for the forward model, so they benefit from surrogate models. Besides that, surrogate models have intrinsic properties to evaluate sensitivity indexes, as they come as a by-product of trained surrogates such as polynomial chaos expansion, low-rank tensor, and random-forest [304] or are evaluated with minor effort in Neural Network [313] and Kriging. Cheng et al. [314] presents an overview of global sensitivity analysis evaluated with surrogate models with their performance comparison. Focusing on improving surrogate interpretability, Pizarroso et al. [315] lists several methods to analyze input-output relationships in ML-based surrogates. Gradient interpretability is one method studied for this purpose [316–318].

Chai et al. [151] constructed a surrogate for the Sound

Transmission Loss analyses of sandwich panels using Random Forest and, as a by-product, got the out-of-bag (OBB) based global sensitivity analysis method. Although the RF surrogate presents bias and smoothing effects, Chai et al. [151] showed an overall good agreement of the sensitivity indexes obtained by the Fourier amplitude sensitivity test (FAST) and OOB-based method and highlighted that the latter can be more easily interpreted. Abbiati et al. [152] show a framework to do global sensitivity analysis in hybrid surrogates, which merge physical and numerical substructures, showing an application in a structural dynamic problem modeled by polynomial chaos expansion surrogates. With a similar method, Abbiati et al. [84] creates a hybrid model for buckling failure reliability analysis using a GP classifier, obtaining a failure surface prediction with good accuracy against experimental and analytical references.

With **Uncertainty Propagation** analysis, the input uncertainties propagate through the model to quantify statistical moments and probability density function (PDF) of the system response, as well as the failure probability [304]. Spectral stochastic methods as Polynomial Chaos Expansion (PCE) and direct simulation methods as Monte Carlo Method (MCM) are used to propagate the uncertainties [156].

Nobari et al. [153] used MCM to quantify the uncertainties of the squeal instability analysis by using a surrogate model of the Complex Eigenvalue Analysis constructed with polynomial regressor and a GP. The model not only provided the PDF of the instability modes, but it also allowed an in-depth knowledge of the effects of each variable on the response, since the global result given by the surrogate diverged from the local sensitivity result obtained with FEM. Hurtado and Alvarez [154] implemented surrogates based on MLP and RBF networks with MCM for analyzing the probability of failure of structures and showed that RBF performed better for the cases under static load, but MLP is better in nonlinear dynamic analysis, where similar inputs may lead to distinct outputs. Wang et al. [155] uses the automatic differentiation property of NN to evaluate the first and second-order derivatives of the surrogate model, to get the response bounds by applying the subinterval method.

Cicirello et al. [146] calls attention to the non-monotonic behavior of vibroacoustic problems, which makes Vertex Methods inappropriate for performing uncertainty propagation of this system. Statistical methods which involve several evaluations of the system are then required. To tackle this problem, Cicirello et al. [146] use Adaptive Sampling techniques with different Acti-

vation Functions to build GP surrogates that predicted the upper and lower bounds of the system. This leads to a decrease in the number of evaluations of the true function during uncertainty propagation. The upper and lower bounds estimations consider the GP variance to lead to more conservative predictions. The proposed method has good accuracy and is faster than a Sub-Interval method, but it struggles when applied to more complex analysis with higher dimensionality. Chapter 8 of [156] reviews traditional uncertainty quantification methods applied to structural dynamics and vibroacoustic problems.

Surrogates may support reliability analysis and risk assessment, in which a failure threshold is established. For nonlinear and time-varying analysis, as often met in vibroacoustic, Extremum Response Surface Method (ERSM) as a surrogate can be handy, since just the extreme responses of the system are considered [157]. Kriging surrogate with moving extremum framework used to model the extreme structural dynamic responses in an interval of time was implemented by Lu et al. [158, 159] to evaluate the reliability and sensitivity analysis of turbine and compressor blisks deformation under dynamic loads. Guo et al. [160] uses active learning Kriging to improve computation efficiency in the reliability analysis of resonance fault of pipelines excited by fluid-structure interactions. Guo et al. [161] used Active Sampling with Kriging to do a sensitivity analysis which quantifies the effects of different variables and the contribution of each failure mode to the system reliability, as for different resonances modes.

The surrogate model implemented by You et al. [162] used Random Forest and Stacking methods to predict the probability of failure of tuned mass damper-based structures under random excitation. The surrogate was trained to replace the entire Monte Carlo analysis and not just the computational model, so each call of the surrogate outputs the probability of failure based on the input distributions.

Specific strategies have been developed for different applications and complexities added to the models. For instance, Bhattacharyya et al. [319] combines Kriging with NARX applied to UQ in dynamical systems since time-domain UQ with Monte Carlo is expensive even when using surrogate models. For high-dimensional problems in which the surrogates may struggle, Tripathy and Bilionis [320] proposes a deep neural network, which comprises an encoder followed by an MLP, to reduce the problem dimension and then apply the uncertainty propagation methods. Luo and Kareem [321] proposes a Deep convolutional neural network approach for dealing with uncertainty quantification in

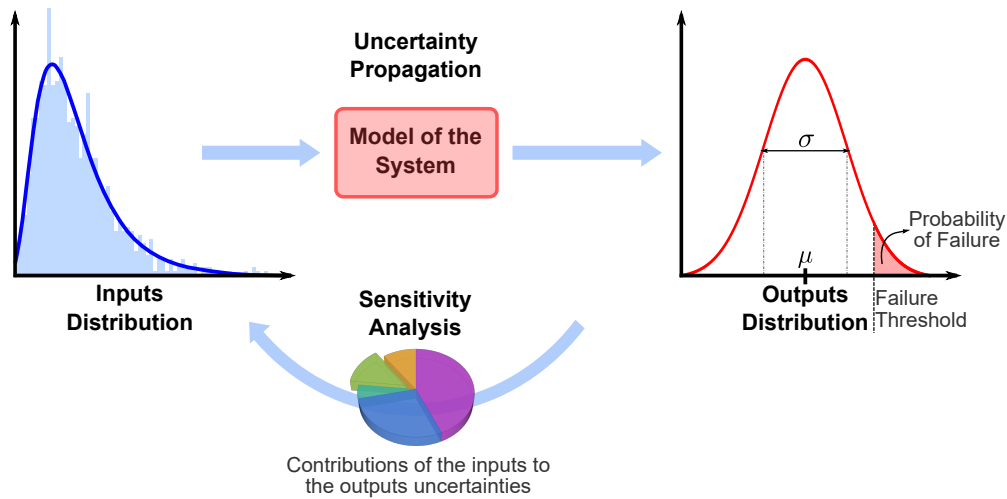


Figure 13: Uncertainty Quantification framework: the Uncertainty Propagation analysis propagates the inputs uncertainties through the system model to obtain the output distribution; and the Sensitivity Analysis evaluates the input contributions to the output uncertainties.

high-dimensional problems, with no dimension reduction. Chaudhuri et al. [322] tackled the complex problem of addressing uncertainty propagation in feedback-coupled multidisciplinary systems using Kriging surrogates and Adaptive Sampling. Böttcher et al. [303] used polymorphic uncertainty propagation method in an eigenvalue problem modeled by Kriging surrogate and Adaptive Sampling.

### 5.3. Optimization with Surrogate Models

Surrogate models are suitable for domain exploration and optimization, being that optimization and multi-objective optimization are the most common links to surrogates in the literature review performed by [299]. Once again, this is due to the surrogate ability to speed up the simulation, allowing to perform several analyses with affordable cost.

Surrogate-based optimization of the vehicle mass subjected to NVH and crashworthiness constraints was performed using an RSM [163] and RBF [164] as surrogate models. The acoustic optimization of an electric motor was tackled by [165] through local surrogates replacing FEM, being that the performance of several ML methods was evaluated, namely Linear regression, Decision Tree, SVM, and GP. Zhang et al. [166] used RBF-based surrogate to replace the modal and vibroacoustic coupling simulation of the volute case of a centrifugal fan in the optimization of the thickness parameters of the geometry to decrease the radiated sound power and the total mass. Transmission Loss (TL) optimization using Gaussian Process surrogates was performed

for intake systems [167] and for meta-material properties [168]. [169] carried out Bandgap optimization of meta-materials supported by RBF surrogate. A surrogate model based on quadratic polynomial regression also leveraged the aerodynamic and acoustic optimization of a fuel cell vehicle fan [323].

Park and Papadimitriou [170] used GP surrogate together with Dynamic substructuring to perform NVH Optimization of a vehicle, while [171] used a surrogate-based in Elman Network, like an RNN, to minimize the vehicle sound pressure while constraining mass, side-impact intrusion, and first-order global modal. Instead of using the vehicle parameters, Tsokaktsidis et al. [172] used time-domain acceleration at the component level as input of an NN surrogate which predicts the sound pressure level in the passenger cabin.

Parametric optimization of the kinematic hardpoints of a vehicle suspension aiming to decrease road noise was performed through an NN surrogate model replacing costly FE analysis [173]. The optimization approach combines criteria that set an FRF curve as an up-limit and as a matching target, aiming to control both amplitudes in a specific frequency and frequency shift. The NN-driven optimization achieved good time-saving and allowed to increase the problem dimensionality in comparison with previous polynomial meta-models constructed for the same problem. Despite that, inaccuracies in the predicted curve are also visible [174]. Inaccuracies do not invalidate surrogates, once they are used in the early stages of the design development and later high-fidelity models and experiments must validate the final results.



Intelligent space exploration methods such as Adaptive Sampling are applied in surrogate-based optimizations as it may be expensive to obtain enough supporting points in the entire domain to build an accurate surrogate. The exploration from Adaptive Sampling allows avoiding local minima, while the exploitation guarantees a good accuracy of the surrogate near the predicted optima. Optimizations using probabilistic surrogates being updated according to the new points selected by the Acquisition Function are called Bayesian optimization and represent a vast field of research.

Back in 1998, Jones et al. [236] published “Efficient Global Optimization (EGO) of Expensive Black-Box Functions”, one of the first studies applying optimization with GPR surrogates. citetFreaan gives an in-depth explanation of Bayesian optimization, while Dwight et al. [305] provides a tutorial with codes for its implementation. Besides that, several toolboxes implement Bayesian Optimization using GP [233, 324, 325], although other probabilistic ML methods are suitable. Mohanasundaram et al. [175] used the EGO approach in the multi-objective optimization of a disc-pad shape under squeal noise criteria modeled by Kriging, after the previous performance of a variance-based sensitivity analysis. Du et al. [176] applied an Adaptive Hierarchical Kriging model to optimize the modal characteristics of an engine. Adaptive Sampling was applied in the optimization of a mechanical low-cutting Metafilter modeled by RBF surrogate [177].

Another use of surrogate models applies to Reliability-Based Design Optimization (RBDO), once both reliability analysis and optimization analysis require several evaluations. Moustapha and Sudret [178] presents a complete survey on surrogate-assisted RBDO with detailed implementation details and several approaches to tackle the reliability analysis. Fei et al. [179] performed an RBDO of turbine blade radial deformation under dynamic loads using an Extreme Support Vector Machine surrogate and Importance Degree Model. Zhang et al. [180] used a fuzzy multi-extremum response surface method to perform an RBDO of fatigue and creep failures of a turbine bladed disk, showing accuracy as in MCM but in a fraction of the time. A reliability EGO approach was implemented to optimize friction-TMD (tuned mass damper) controlled structures modeled by Kriging [181]. Polynomial-Chaos-based Kriging was used as a surrogate to speed up the dynamic simulations in the RBDO of a Nonlinear Energy Sink, a passive control device to mitigate vibration [326]. In his Thesis, Moustapha [327] did a RBDO using GPR and Adaptive Sampling in the buckling analysis of a column

under pressure aiming to minimize its cross area while keeping a constraint in the probability of failure. The accuracy criterion used in the Adaptive Sampling algorithm decreases with optimization iterations to guarantee the surrogate accuracy near the optimal point. Moustapha also applied this methodology to a complex real-life application: the RBDO of the crash analysis of a lightweight vehicle.

Besides adaptive sampling, global approximations with local refinements and domain-decomposition-based methods are other techniques to improve the surrogates in the region of interest, as regions near the optimal or with highly localized behavior [300]. Importance sampling is also performed to decrease the number of evaluation points in uncertainty propagation analysis [328].

Neural Network-based surrogate model could benefit from the automatic differentiation (AD) [196] properties of NN to perform optimization by using the derivative of the output function. A computational packet with AD implemented is shown by Bouhlel et al. [329], as well as an example of optimization implementation using AD. However, AD is difficult to implement for complex NN architectures.

## 6. On Future Trends and Perspectives

Digital transformation is already a reality and has been changing the way to solve several problems, including mechanical problems traditionally solved solely by physical models. The various works referenced in this article illustrate how this transformation is taking place and bringing advantages to acoustic and structural dynamic field. Despite the current progress, much should be done to scale DT implementation and to take full advantage of them.

**Integration** is a cornerstone on this path and two main discussions field are raising in this direction: **Digital Twins**, which approach the concept of integrating all levels of simulation and information of an asset through its life-cycle; and **Physics Guided Machine Learning**, in which physics knowledge is embedded into data-driven methods to support learning of consistent representations. Next, these and other topics are discussed to evaluate future paths in ML research applied to vibroacoustic problems through the observation of recent results of ML in other physic domains and through the observation of the current gaps in the field.

### 6.1. Digital Twin

The Digital Twin is a time-evolving highly fidelity replica of a product/process, with a bidirectional con-

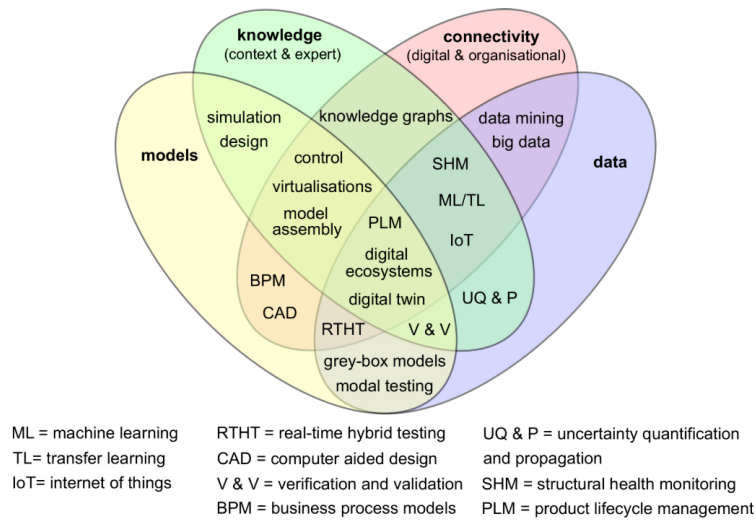


Figure 14: Main components and interconnections which are the building blocks for creating a Digital Twin. Source: Gardner et al. [312].

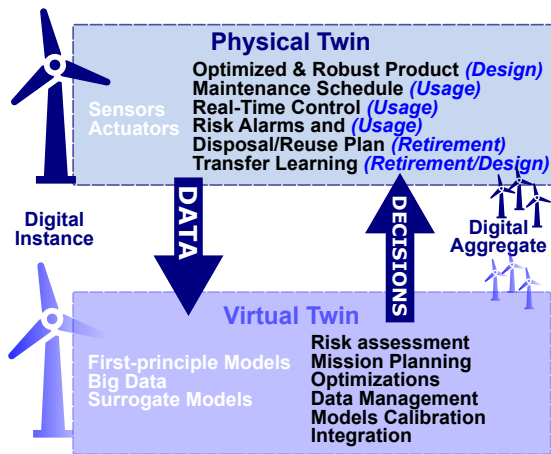


Figure 15: Digital Twin framework: the data from the Physical Twin is processed by Physics and Data Driven methods by the Digital Twin, which supports optimized and robust decisions throughout the product life cycle.

nection of information. The concept was first addressed in 2003 by Grieves in a presentation on Product Lifecycle Management (PLM) [330], but it only spread with the famous article by Glaessgen and Stargel [331]. This article envisioned the great potential of DT as an integrated multiphysics and multiscale simulation of the real system using the best physical models and data available, to create a virtual copy able to continuously forecast the system health and create plans to mitigate the damage or improve performance while accounting for the system associated uncertainties.

However, the concept of DT is still loose and broad and is constantly evolving while DT enablers are under ongoing development and DT applications are spreading to many sectors. In view of this, several works focused on reviewing the characteristics and achievements of the DT [85, 302, 332–344].

According to Gardner et al. [312], the Digital Twin is built from components from four main categories: **simulations** which model the physics of the system; the **knowledge** from experts and previous experiences about the product and the environment variables; the available **data** of the physical twin; and the **connectivity** which links the other elements and gives DT the ability to evolve with information. These components and their interconnections are the building blocks for creating a Digital Twin, as illustrated in Figure 14.

As pointed out by many authors [330, 332, 337, 339, 343, 345–347], the DT must evolve throughout the life of the product. During the product development, where DT is called Digital Prototype [332], surrogate models

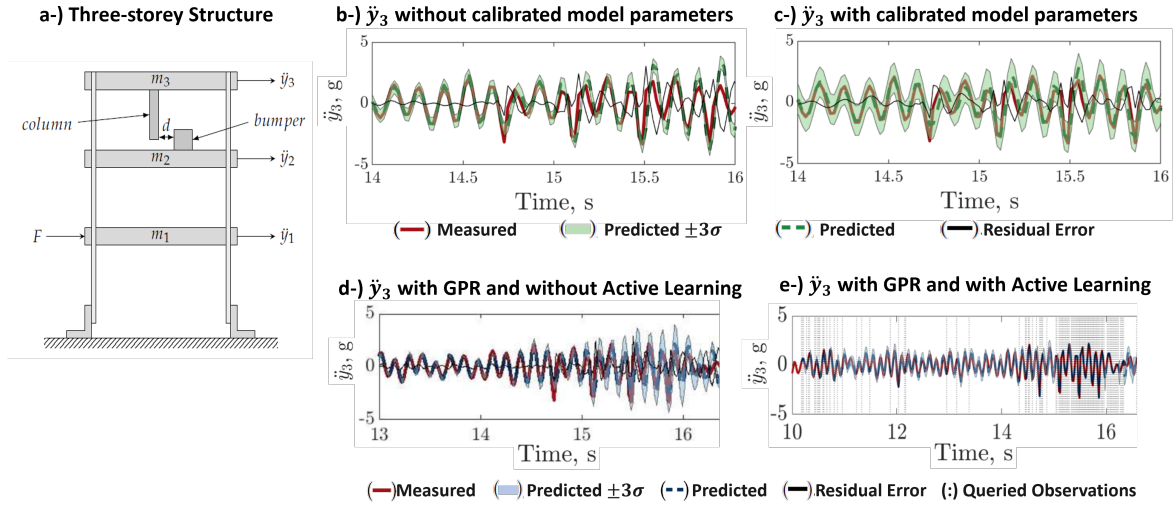


Figure 16: Three-storey structure used by Gardner et al. [312] to construct an operational DT (a) and predicted acceleration response of the third floor  $\ddot{y}_3$  when column and bumper are in contact (nonlinear response) for different stages of the DT implementation (b-e). Adapted from [312].

are used to explore the design space, leading to optimized and robust design. During the usage phase, monitoring the product and its environment assists in the early detection of failures, optimization of control strategies and mission planning. Data can also be processed and merged to generate virtual sensors, leading to more informative operation without extra hardware [348, 349]. Finally, component life estimation is used to optimize scheduled maintenance and to support end-of-life decisions on disposal, reuse and market value [350]. A representation of the Digital Twin components, advantages and uses throughout the life cycle is shown in Figure 15.

The information collected throughout the product life cycle also helps in the design of the next generation, as it is possible to evaluate the components that were over- or under-designed. Moreover, DT could make it possible to investigate the causality of the observed phenomena by exploring sensitivity features in Section 5.2. Therefore, a complete DT must store and manage the product data, as well as integrate data-driven and high-fidelity simulations, both for an individual product (Digital Instance) and an assembly of them (Digital Aggregate) [332]. In summary, the DT aims to avoid wasting valuable data and information.

A complete DT does not exist yet, and its implementation might take decades of further development, as predicted by Glaessgen and Stargel [331]. However, integrating several key elements has led to the development of incomplete DTs that are noteworthy. One example aforementioned is the DT applied to SHM de-

veloped by Karve et al. [95], which merges data and physical-driven methods to perform a mission-planning that minimizes damage while accounting for uncertainties, meaning that the DT is time-evolving and has a bi-directional connection.

The DT of an aircraft implemented by Kapteyn et al. [96] points to an interesting route. The DT identifies the current damage scenario through a classification method and selects the proper surrogate model, making an informed decision to replan the maneuvers. Aivaliotis et al. [349] presents a methodology of DT implementation in predictive maintenance, including physics-based modeling, virtual sensors modeling, automatic calibration of model parameters. The implemented DT is used to deliver RUL predictions, as demonstrated in the case study of an industrial welding robot.

Gardner et al. [312] implemented a DT in several stages. First, measured data was used to calibrate the physical model parameters. Then the outputs of this model are used as input of a GPR, which ameliorates the output prediction to care for uncertainties and non-modeled physics using online adaptive sampling. The methodology is demonstrated in the model of the three-storey structure, as shown in Figure 16. Although the physical model used was linear, the DT could predict the nonlinear behavior resulting from the contact between column and bumper, which occurs at specific excitations. Besides that, as the DT is trained with lagged information and can make predictions steps ahead in time, it is conveniently used in the Active Control of the structure.

Table 1: Advantages of Physics-Guided Machine Learning distinguishing the contributions of the data-driven approach and of the embedded physics knowledge.

Advantages of PGML provided by the data-driven model	Advantages of PGML provided by the physics embedded on it
Improve state-of-the-art physical models by comprising unknown relations; Solve inverse problems and lead to better parameter identification in the physical model; Handle noisy input; Reduce model order; Estimate aleatory and epistemic errors bounds; Mitigate instability issues in time integrators; Provide lagged predictions to active control; Discover governing equations and unknown physics; Computationally cheap to evaluate.	Improve ML predictions with domain knowledge and inductive bias; Provide physically consistent models; Reduce or end need of data; Increase interpretability of ML model; Improve ML generalization for unseen scenarios; Reduce search space of ML algorithm Improve long term-forecasting.

## 6.2. Physics Guided Machine Learning

The big drawback of ML models, especially when applied to physical problems, is the lack of the theoretical base and interpretability, raising skepticism with ML by part of the scientific community. Indeed, ML models may lead to physically inconsistent results, may fail to generalize to unseen scenarios, and rely on the availability of big data. However, Physical-driven models (PDM) rely on hypotheses and simplifications of the real boundary conditions and struggle to account for uncertainties and historical and environmental conditions. Physics Guided Machine Learning (PGML) is an incipient but fast-growing research field which suggest merging physics-driven and data-driven model to take the best of both worlds, as shown in Table 1 [302, 351–354].

Recent reviews by Willard et al. [302], Karpatne et al. [351], Rai and Sahu [355] and Wang and Yu [352] classify and describe PGML works developed in different domains in the last years. In his thesis, Stender [356] develops a data science process for mechanical vibrations explicitly considering physics aspects in all steps of the process, namely obtain, pre-process, transform, model, and explain (OPTME). Some of the ML applied in SHM, Active Control, and surrogates from the last sections might be classified as PGML.

The state-of-art of PGML is described here according to the configurations in which the physical knowledge is merged with the ML algorithm, as illustrated in Figure 17. Two categories can be defined: **Physics leveraged by Machine Learning**, in which ML models are used to improve the results from the simplified physical

models; and **Machine Learning leveraged by Physics**, in which physical laws and constraints are intrinsically embedded into the ML, guiding it to have physical consistent results. Willard et al. [302] presented a similar categorization and provided a table associating each PGML configuration with an objective for which it may be appropriate.

One way in which ML can leverage the results of physical simulations is when the results of the latter (and possibly its inputs) are used as ML input in a **In-series hybrid model** configuration. The ML is trained to correct the results of the physical model by using the output of the real system as the target [312, 357]. Similarly, in **Residual Modelling**, ML learns to model the PDM error, therefore the ML can correct the PDM output or classify its validity, as in [353, 358–362]. Finally, the ML can be used just as a sub-process of the PDM to evaluate one of its parameters [95, 363–367].

In the configurations that physics leverage ML, the structure is case-specific since it depends on the physical equations which govern the system. The most common approach is **Physics Guided Loss** [357, 368–374], in which the loss function contains penalization terms for non-physical predictions, e.g., an unexpected non-monotonic behavior. A thorough case of Physics Guided Loss is in Physics-Informed Neural Network (PINN) [375–385], in which the loss function is solely composed of the residue of a partial differential equation formulated in its derivative form. The equation variables are also the NN inputs, therefore the residue (loss function) is minimized by using automatic differentia-

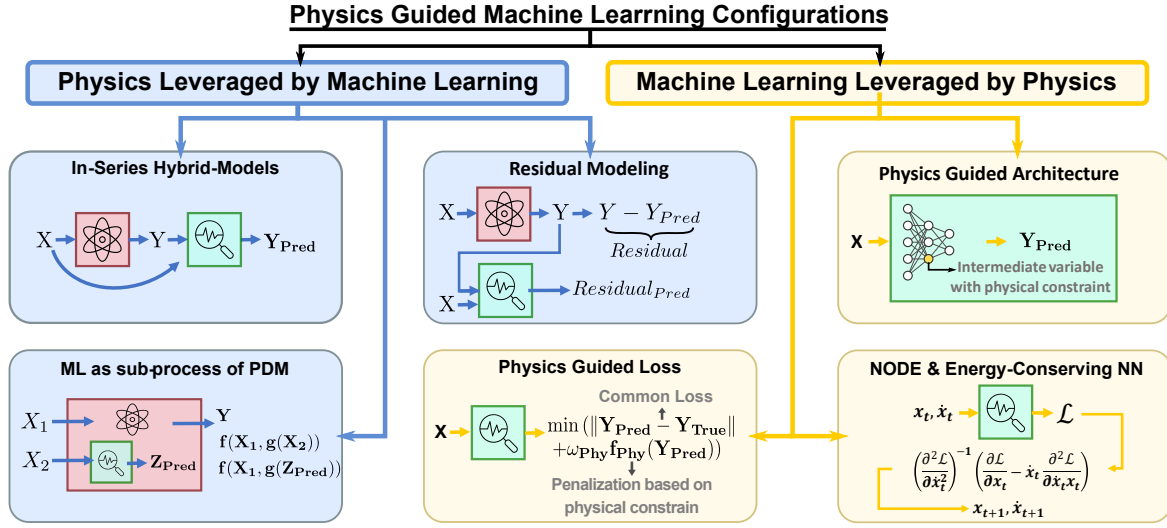


Figure 17: Configurations of Physics-Guided Machine Learning merging Data-Driven Models (DDM) and Physics-Driven Models (PDM). The representations are illustrative and do not show the arrangement precisely nor include all configurations.

tion of NNs [386] and the equation is solved with no data needed. PINN also solves inverse problems, discovering equation parameters or constitutive relationships [387–396].

Another popular approach is **Physics-guided architecture**, in which a physical behavior is inserted somehow in the model architecture. At some level, this is done to insert sequential behavior in RNN, for example. Expected physical behavior can be embedded through constraints in weights and biases [397] or intermediate variables [371, 398].

Zhang et al. [211] used LSTM and graph-based tensor differentiator to enforce physical constraints in the architecture and loss-function of metamodels of nonlinear structural systems. Besides improving the prediction accuracy and robustness, the PGML implemented in [211] models non-observable latent nonlinear state variables, such as the hysteretic meter, and nonlinear restoring force, delivering a more interpretable surrogate.

Elements of physics-guided loss and physics-guided architecture are used in **Neural ordinary differential equations (NODE)** and **Energy-Conserving Neural Networks (ECNN)**. In NODEs, explicit integration steps are performed in each layer of the NN as one step evaluation of a common ODE solver [399–403]. In ECNN, the structure of Lagrangian and Hamiltonian equations have been embedded into the NN construction to ensure an energy conservative behavior, as reviewed by Lutter and Peters [404] and implemented in different structures in [399, 405–413].

Willard et al. [302] survey presents other approaches, while Ba et al. [414] merged several PGML approaches to create an NN able to generalize well to different mechanical problems. Although it is a new topic, several recent works employed PGML, underlying its potential. However, most PGML research concentrate in other fields, e.g., fluids dynamics [22, 359, 415], lake modeling [357, 368, 369, 371, 416], climate modelling [417, 418] and material science [21, 419, 420]. The spread of PGML techniques to other domains is a matter of time.

For example, NODE and ECNN can be used as time integration solvers for modeling structural dynamics and acoustics and the physics information can help to solve the difficulties posed by the rough and discontinuous behavior of these systems. Recently, [421] created a PGML of a structural dynamic system using an RNN encoding the equation of motion. The PGML showed superior results even for scarce and noisy data, with better generalizability and robustness compared to purely data-driven. Besides that, it allowed time-saving by applying big time-steps without facing stability issues from the purely mechanistic approach. NODE and ECNN are suitable for introducing inductive biases in dynamic systems. Examples of how this approach can improve ML performance under high nonlinearities and discontinuities are presented in recent works which applied ECNNs to leverage data efficiency in non-smooth contact dynamics problems [214, 422].

Yin et al. [423] introduced the APHYNITY framework to augment physical models with data information

applied to dynamics forecasting. The Residual Modeling approach considers that the final response constitutes in both physics and ML models while ensuring that the ML response contribution is minimal so that the physics explains most of the prediction as possible. In addition, ECNNs are used in the APHYNITY framework to ensure physical consistency. The work presents results for reaction-diffusion equations, wave equations, and nonlinear damped pendulum, showing better accuracy than the simplified physical-based model and the solely data-driven approach. APHYNITY also leveraged the identification of physical parameters.

Thus, PGML could ameliorate ML techniques used in the applications mentioned in this paper, increasing the coherence, interpretability, and reliability of ML models in structural dynamics and acoustic. Besides that, a burgeoning discussion explores how to use PGML to unveil unknown governing equations and physics intuition based on data [293, 294, 369, 424–426]. Recently, Lai et al. [427] applied NODE to learn the governing structural dynamics and experimentally showed its effectiveness in a structure equipped with a negative stiffness device. Incipient research applied ECNN to learn the dynamics of the pendulum and multi-body problems [399, 405, 428, 429]. Further research in the area might consider dynamic system with flexible elements.

### 6.3. Research gaps and emerging opportunities

This survey identified drawbacks and difficulties in the employment of ML in SD&V that should be addressed in future works. Based on the spotted research gaps and in the observed research trends in the integration of ML with other physical sciences, some future research opportunities that arise are:

- Explore the many configurations of Physics-Guided ML in vibroacoustic problems to enforce physics consistency and improve accuracy, as carried out in other physical domains by [211, 404].
- Ameliorate ML interpretability with sensitivity analysis [162] and PGML [211].
- Use ML to discover governing equations [293, 294, 369, 424–426] and meta-materials [21, 419] in SD&V problems.
- Create a Digital Twin in the vibroacoustic domain by using entire life-cycle data, integrating multiple assets information, and performing real-time decision-making.

- Investigate lifelong learning [430] applied to SHM, Active Control, and Product Design.
- Overcome problems due to the lack of a dedicated database in SHM through transfer learning methods, similitude methods applied to SHM signals, and crowd-sourcing database [247].
- Use trained ML models to identify and mitigate the cause of failures.
- The research exploring surrogate models in vibroacoustic problems is still incipient. To overcome the difficulties because of the rough behavior of the functions, one should investigate approaches such as learning from the function derivatives [431], global surrogates with local refinements, domain-decomposition methods [300], Adaptive Sampling [301], and Physics-Guided Machine Learning methods [302].
- Present a systematic study to clarify in which scenarios a surrogate model is justifiable in SD&V applications, considering the problem dimensionality, function smoothness, number of supporting points, loss of accuracy, and time gains.
- Study methods to improve robustness of ML-based control of noise and vibration, as adversarial reward learning with reinforcement learning [432], NN with provable guarantees [433], and many others presented in [434].

## 7. Conclusion and Discussion

This article presents a review of the intersection between the fields of Machine Learning (ML) and Structural Dynamics and Vibroacoustic (SD&V). First, the main ML methods are revised, paving the way for a broader and more advanced understanding of ML tools applied in the field. Subsequently, the reviewed literature reveals the ability of ML to perform critical tasks in SD&V, often surpassing physics-based methods in efficiency and accuracy. ML is notable for handling nonlinear analyses, uncertainties, and expensive/unmodeled physical problems. Three major application areas were identified: Structural Health Monitoring (SHM), Active Control of noise and vibration, and vibroacoustic Product Design.

The capabilities of ML in extracting and recognizing fault patterns from monitored signals in the time and frequency domain make SHM the most developed and explored of these areas. Recurrently used data transformation and preprocessing methods are presented with

examples of their significance in prediction accuracy. The prominence of Deep-learning methods in SHM is noteworthy since they automatically extract relevant features and reveal more complex patterns. SHM allows early failure detection and enables Remaining Useful Life predictions. Consequently, SHM avoids catastrophic failures and might implement a preventive maintenance schedule to optimize uptime, maximizing the use of component lifetime. In addition, ML methods locate and classify damage and plan actions to minimize risks. Therefore, ML for SHM results in operational safety and economic savings, with simpler implementations and better efficiency than model-based methods.

The correspondence between the areas of ML and Active Control of Vibration and Noise is natural since both have data-based optimization as a pillar. Thus, active control theory exploits ML tools by various means. System identification and reduced-order modeling techniques use ML to model the system under control. ML methods also support the study of the optimal location of sensors and actuators. Various control system approaches and configurations use ML methods to optimize controller parameters and policies, offline or online. Usually, ML applications in Active Control are justified and show better performance than traditional methods in nonlinear and complex scenarios. Although the memory and processing costs of ML algorithms are prohibitive in many control applications, the computer science advances occurring in parallel should make them affordable in more scenarios.

Replacing costly high-fidelity simulations with fast ML-based surrogate models also enhances product design. The alleviation of the computational cost enables the evaluation of more designs, allowing uncertainty propagation and optimization analyses. Moreover, the statistical ground of ML methods makes them suitable to handle uncertainties, being that many have built-in sensitivity analysis to improve the comprehension of the system physics and ML interpretability. Besides that, ML surrogates trained with adaptive sampling require fewer observation points, while ensuring better accuracy near regions of interest, being widely used in optimizations. Despite the advantages, surrogates are not used in SD&V as often as in other domains. This occurs mainly because of the difficulties found by ML when fitting functions with rough behavior, recurrent in SD&V. However, methods such as domain subdivision, adaptive sampling, local refinements, and ML guided by physics are areas of potential study to overcome this issue.

The wide range of ML application possibilities in SD&V and the advantages of this integration are ex-

PLICIT in this article, justifying the great interest in the area. Furthermore, the advances of successful recent research, likewise the analysis of ML uses in other fields of natural science, point to a path of opportunities. Connectivity and data management are the key aspects explored in the Digital Twins concept, which aims to benefit from the data and knowledge available to improve Product Lifecycle Management. Incorporating more theoretical and expert knowledge into ML models, as studied in Physics Guided Machine Learning models, is another trend subject of research, onde it leads to more interpretable models, less need for training data, and more physically consistent predictions. The drawbacks and difficulties identified in the application of ML in SHM, Active Control and Product design also show the open discussions and room for progress. The authors' perspectives for the upcoming research fields merging ML and SD&V were compiled in a list to be used as a guideline for future research.

The extensive review presented reinforces that ML can strongly collaborate for the development of SD&V projects that are safer, more stable, controllable, robust, and optimized in design and operation. In conclusion, the union of data-driven and physics-driven methods can lead to a greater understanding of phenomena involved in SD&V analyses and open the way for further developments in the field.

## 8. Acknowledgements

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 860243. The author would like to acknowledge all the Institutions and Partners involved within the LIVE-I project.

## References

- [1] M. Hilbert, Digital technology and social change: The digital transformation of society from a historical perspective, *Dialogues in Clinical Neuroscience* 22 (2020) 189.
- [2] K.-H. Yu, A. L. Beam, I. S. Kohane, Artificial intelligence in healthcare, *Nature biomedical engineering* 2 (2018) 719–731.
- [3] P. Mallozzi, P. Pelliccione, A. Knauss, C. Berger, N. Mohammadiha, Autonomous vehicles: state of the art, future trends, and challenges, *Automotive Systems and Software Engineering* (2019) 347–367.
- [4] M. C. Medeiros, G. F. Vasconcelos, Á. Veiga, E. Zilberman, Forecasting inflation in a data-rich environment: the benefits of machine learning methods, *Journal of Business & Economic Statistics* 39 (2021) 98–119.
- [5] A. A. A. Ahmed, A. Aljabouh, P. K. Donepudi, M. S. Choi, Detecting fake news using machine learning: A systematic literature review, arXiv preprint arXiv:2102.04458 (2021).

- [6] Z. Gharibshah, X. Zhu, User response prediction in online advertising, *ACM Computing Surveys (CSUR)* 54 (2021) 1–43.
- [7] T. Balaji, C. S. R. Annavarapu, A. Bablani, Machine learning algorithms for social media analysis: A survey, *Computer Science Review* 40 (2021) 100395.
- [8] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al., Mastering the game of go with deep neural networks and tree search, *nature* 529 (2016) 484–489.
- [9] A. Voulodimos, N. Doulamis, A. Doulamis, E. Protopapadakis, Deep learning for computer vision: A brief review, *Computational intelligence and neuroscience* 2018 (2018).
- [10] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, I. Sutskever, Zero-shot text-to-image generation, 2021. [arXiv:2102.12092](https://arxiv.org/abs/2102.12092).
- [11] J. Hirschberg, C. D. Manning, Advances in natural language processing, *Science* 349 (2015) 261–266.
- [12] M. Frank, D. Drikakis, V. Charissis, Machine-learning methods for computational science and engineering, *Computation* 8 (2020) 15.
- [13] J. G. Greener, S. M. Kandathil, L. Moffat, D. T. Jones, A guide to machine learning for biologists, *Nature Reviews Molecular Cell Biology* 23 (2022) 40–55.
- [14] N. Artrith, K. T. Butler, F.-X. Coudert, S. Han, O. Isayev, A. Jain, A. Walsh, Best practices in machine learning for chemistry, *Nature Chemistry* 13 (2021) 505–508.
- [15] J. P. Janet, H. J. Kulik, *Machine Learning in Chemistry*, American Chemical Society, 2020.
- [16] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, L. Zdeborová, Machine learning and the physical sciences, *Reviews of Modern Physics* 91 (2019) 045002.
- [17] P. Mehta, M. Bukov, C.-H. Wang, A. G. Day, C. Richardson, C. K. Fisher, D. J. Schwab, A high-bias, low-variance introduction to machine learning for physicists, *Physics reports* 810 (2019) 1–124.
- [18] M. Feickert, B. Nachman, A living review of machine learning for particle physics, [arXiv preprint arXiv:2102.02770](https://arxiv.org/abs/2102.02770) (2021).
- [19] A. Radovic, M. Williams, D. Rousseau, M. Kagan, D. Bonaccorsi, A. Himmel, A. Aurisano, K. Terao, T. Wongjirad, Machine learning at the energy and intensity frontiers of particle physics, *Nature* 560 (2018) 41–48.
- [20] K. T. Butler, D. W. Davies, H. Cartwright, O. Isayev, A. Walsh, Machine learning for molecular and materials science, *Nature* 559 (2018) 547–555.
- [21] J. Schmidt, M. R. Marques, S. Botti, M. A. Marques, Recent advances and applications of machine learning in solid-state materials science, *npj Computational Materials* 5 (2019) 1–36.
- [22] S. L. Brunton, B. R. Noack, P. Koumoutsakos, Machine learning for fluid mechanics, *Annual Review of Fluid Mechanics* 52 (2020) 477–508.
- [23] Y. Xie, M. Ebad Sichani, J. E. Padgett, R. DesRoches, The promise of implementing machine learning in earthquake engineering: A state-of-the-art review, *Earthquake Spectra* 36 (2020) 1769–1801.
- [24] Q. Kong, D. T. Trugman, Z. E. Ross, M. J. Bianco, B. J. Meade, P. Gerstoft, Machine learning in seismology: Turning data into insights, *Seismological Research Letters* 90 (2019) 3–14.
- [25] M. H. Ahmadi, R. Kumar, M. E. H. Assad, P. T. T. Ngo, Applications of machine learning methods in modeling various types of heat pipes: a review, *Journal of Thermal Analysis and Calorimetry* (2021) 1–9.
- [26] X. Qian, R. Yang, Machine learning for predicting thermal transport properties of solids, *Materials Science and Engineering: R: Reports* 146 (2021) 100642.
- [27] A. Mosavi, M. Salimi, S. Faizollahzadeh Ardabili, T. Rabczuk, S. Shamshirband, A. R. Varkonyi-Koczy, State of the art of machine learning models in energy systems, a systematic review, *Energies* 12 (2019) 1301.
- [28] C. R. Farrar, K. Worden, *Structural health monitoring: a machine learning perspective*, John Wiley & Sons, 2012.
- [29] S. Khan, T. Yairi, A review on the application of deep learning in system health management, *Mechanical Systems and Signal Processing* 107 (2018) 241–265.
- [30] 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 (2020) 2778.
- [31] G. Toh, J. Park, Review of vibration-based structural health monitoring using deep learning, *Applied Sciences* 10 (2020) 1680.
- [32] Y.-z. Lin, Z.-h. Nie, H.-w. Ma, Structural damage detection with automatic feature-extraction through deep learning, *Computer-Aided Civil and Infrastructure Engineering* 32 (2017) 1025–1046.
- [33] Y. Bao, H. Li, Machine learning paradigm for structural health monitoring, *Structural Health Monitoring* 20 (2021) 1353–1372.
- [34] A. Malekloo, E. Ozer, M. AlHamaydeh, M. Girolami, Machine learning and structural health monitoring overview with emerging technology and high-dimensional data source highlights, *Structural Health Monitoring* (2021) 14759217211036880.
- [35] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, R. X. Gao, Deep learning and its applications to machine health monitoring, *Mechanical Systems and Signal Processing* 115 (2019) 213–237.
- [36] R. Liu, B. Yang, E. Zio, X. Chen, Artificial intelligence for fault diagnosis of rotating machinery: A review, *Mechanical Systems and Signal Processing* 108 (2018) 33–47.
- [37] A. Heng, S. Zhang, A. C. Tan, J. Mathew, Rotating machinery prognostics: State of the art, challenges and opportunities, *Mechanical systems and signal processing* 23 (2009) 724–739.
- [38] M. Flah, I. Nunez, W. B. Chaabene, M. L. Nehdi, Machine learning algorithms in civil structural health monitoring: A systematic review, *Archives of Computational Methods in Engineering* 28 (2021) 2621–2643.
- [39] O. Avci, O. Abdeljaber, S. Kiranyaz, M. Hussein, M. Gabbouj, D. J. Inman, A review of vibration-based damage detection in civil structures: From traditional methods to machine learning and deep learning applications, *Mechanical systems and signal processing* 147 (2021) 107077.
- [40] R. Hou, Y. Xia, Review on the new development of vibration-based damage identification for civil engineering structures: 2010–2019, *Journal of Sound and Vibration* 491 (2021) 115741.
- [41] X. W. Ye, T. Jin, C. B. Yun, A review on deep learning-based structural health monitoring of civil infrastructures, *Smart Structures and Systems* 24 (2019) 567–585. URL: [www.scopus.com](https://www.scopus.com), cited By :73.
- [42] J. Yang, W. Zhang, Y. Liu, Subcycle fatigue crack growth mechanism investigation for aluminum alloys and steel, in: 54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 2013, p. 1499.
- [43] H.-K. Wang, R. Haynes, H.-Z. Huang, L. Dong, S. N. Atluri, The use of high-performance fatigue mechanics and the extended kalman/particle filters, for diagnostics and prognostics of aircraft structures, *CMES: Computer Modeling in Engineering & Sciences* 105 (2015) 1–24.
- [44] B. R. Seshadri, T. Krishnamurthy, *Structural health manage-*



- ment of damaged aircraft structures using digital twin concept, in: 25th aiaa/ahs adaptive structures conference, 2017, p. 1675.
- [45] A. J. Zakrajsek, S. Mall, The development and use of a digital twin model for tire touchdown health monitoring, in: 58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 2017, p. 0863. doi:<https://doi.org/10.2514/6.2017-0863>.
- [46] L. Janssen, I. L. Arteaga, Data processing and augmentation of acoustic array signals for fault detection with machine learning, *Journal of Sound and Vibration* 483 (2020) 115483.
- [47] O. Gecgel, S. Ekwaro-Osire, J. P. Dias, A. Serwadda, F. M. Alemayehu, A. Nispel, Gearbox fault diagnostics using deep learning with simulated data, in: 2019 IEEE International Conference on Prognostics and Health Management (ICPHM), IEEE, 2019, pp. 1–8.
- [48] H. C. Choe, Y. Wan, A. K. Chan, Neural pattern identification of railroad wheel-bearing faults from audible acoustic signals: comparison of fft, cwt, and dwt features, in: *Wavelet Applications IV*, volume 3078, International Society for Optics and Photonics, 1997, pp. 480–496.
- [49] M. R. Taha, A. Noureldin, J. Lucero, T. Baca, Wavelet transform for structural health monitoring: a compendium of uses and features, *Structural health monitoring* 5 (2006) 267–295.
- [50] T. Junbo, L. Weining, A. Juneng, W. Xueqian, Fault diagnosis method study in roller bearing based on wavelet transform and stacked auto-encoder, in: The 27th Chinese Control and Decision Conference (2015 CCDC), IEEE, 2015, pp. 4608–4613.
- [51] S. Khan, P. Phillips, I. Jennions, C. Hockley, No fault found events in maintenance engineering part I: Current trends, implications and organizational practices, *Reliability Engineering & System Safety* 123 (2014) 183–195.
- [52] P. Shi, X. Guo, D. Han, R. Fu, A sparse auto-encoder method based on compressed sensing and wavelet packet energy entropy for rolling bearing intelligent fault diagnosis., *Journal of Mechanical Science & Technology* 34 (2020).
- [53] R. Thirukovalluru, S. Dixit, R. K. Sevakula, N. K. Verma, A. Salour, Generating feature sets for fault diagnosis using denoising stacked auto-encoder, in: 2016 IEEE International Conference on Prognostics and Health Management (ICPHM), IEEE, 2016, pp. 1–7.
- [54] C. Li, R.-V. Sanchez, G. Zurita, M. Cerrada, D. Cabrera, R. E. Vásquez, Gearbox fault diagnosis based on deep random forest fusion of acoustic and vibratory signals, *Mechanical Systems and Signal Processing* 76 (2016) 283–293.
- [55] J. Wang, J. Zhuang, L. Duan, W. Cheng, A multi-scale convolution neural network for featureless fault diagnosis, in: 2016 International Symposium on Flexible Automation (ISFA), IEEE, 2016, pp. 65–70.
- [56] H. Liu, L. Li, J. Ma, Rolling bearing fault diagnosis based on stft-deep learning and sound signals, *Shock and Vibration* 2016 (2016).
- [57] D. Verstraete, A. Ferrada, E. L. Droguett, V. Meruane, M. Modarres, Deep learning enabled fault diagnosis using time-frequency image analysis of rolling element bearings, *Shock and Vibration* 2017 (2017).
- [58] S. K. Singh, S. Kumar, J. Dwivedi, Compound fault prediction of rolling bearing using multimedia data, *Multimedia Tools and Applications* 76 (2017) 18771–18788.
- [59] X. Guo, L. Chen, C. Shen, Hierarchical adaptive deep convolution neural network and its application to bearing fault diagnosis, *Measurement* 93 (2016) 490–502.
- [60] H. Oh, B. C. Jeon, J. H. Jung, B. D. Youn, Smart diagnosis of journal bearing rotor systems: Unsupervised feature extraction scheme by deep learning, in: Annual Conference of the PHM Society, 1, 2016, pp. 1–8.
- [61] S. Li, G. Liu, X. Tang, J. Lu, J. Hu, An ensemble deep convolutional neural network model with improved ds evidence fusion for bearing fault diagnosis, *Sensors* 17 (2017) 1729.
- [62] O. Janssens, V. Slavkovikj, B. Vervisch, K. Stockman, M. Locufier, S. Verstockt, R. Van de Walle, S. Van Hoecke, Convolutional neural network based fault detection for rotating machinery, *Journal of Sound and Vibration* 377 (2016) 331–345.
- [63] G. S. Babu, P. Zhao, X.-L. Li, Deep convolutional neural network based regression approach for estimation of remaining useful life, in: International conference on database systems for advanced applications, Springer, 2016, pp. 214–228.
- [64] P. Wang, R. Yan, R. X. Gao, et al., Virtualization and deep recognition for system fault classification, *Journal of Manufacturing Systems* 44 (2017) 310–316.
- [65] W. Sun, B. Yao, N. Zeng, B. Chen, Y. He, X. Cao, W. He, An intelligent gear fault diagnosis methodology using a complex wavelet enhanced convolutional neural network, *Materials* 10 (2017) 790.
- [66] T. Ince, S. Kiranyaz, L. Eren, M. Askar, M. Gabbouj, Real-time motor fault detection by 1-d convolutional neural networks, *IEEE Transactions on Industrial Electronics* 63 (2016) 7067–7075.
- [67] O. Abdeljaber, O. Avci, S. Kiranyaz, M. Gabbouj, D. J. Inman, Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks, *Journal of Sound and Vibration* 388 (2017) 154–170.
- [68] L. Jing, M. Zhao, P. Li, X. Xu, A convolutional neural network based feature learning and fault diagnosis method for the condition monitoring of gearbox, *Measurement* 111 (2017) 1–10.
- [69] W. Zhang, C. Li, G. Peng, Y. Chen, Z. Zhang, A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working load, *Mechanical Systems and Signal Processing* 100 (2018) 439–453.
- [70] W. Zhang, G. Peng, C. Li, Y. Chen, Z. Zhang, A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals, *Sensors* 17 (2017) 425.
- [71] D. Cabrera, F. Sancho, C. Li, M. Cerrada, R.-V. Sánchez, F. Pacheco, J. V. de Oliveira, Automatic feature extraction of time-series applied to fault severity assessment of helical gearbox in stationary and non-stationary speed operation, *Applied Soft Computing* 58 (2017) 53–64.
- [72] W. Sun, R. Zhao, R. Yan, S. Shao, X. Chen, Convolutional discriminative feature learning for induction motor fault diagnosis, *IEEE Transactions on Industrial Informatics* 13 (2017) 1350–1359.
- [73] Z. Chen, C. Li, R.-V. Sanchez, Gearbox fault identification and classification with convolutional neural networks, *Shock and Vibration* 2015 (2015).
- [74] J. Tao, Y. Liu, D. Yang, Bearing fault diagnosis based on deep belief network and multisensor information fusion, *Shock and Vibration* 2016 (2016).
- [75] Z. Chen, C. Li, R.-V. Sánchez, Multi-layer neural network with deep belief network for gearbox fault diagnosis, *Journal of Vibroengineering* 17 (2015) 2379–2392.
- [76] R. Zhao, R. Yan, J. Wang, K. Mao, Learning to monitor machine health with convolutional bi-directional lstm networks, *Sensors* 17 (2017) 273.
- [77] G. Fan, J. Li, H. Hao, Vibration signal denoising for structural health monitoring by residual convolutional neural networks, *Measurement* 157 (2020) 107651.
- [78] P. Rizzo, M. Cammarata, D. Dutta, H. Sohn, K. Harries, An unsupervised learning algorithm for fatigue crack detection in waveguides, *Smart Materials and Structures* 18 (2009) 025016.

- [79] K. K. Reddy, S. Sarkar, V. Venugopalan, M. Giering, Anomaly detection and fault disambiguation in large flight data: a multi-modal deep auto-encoder approach, in: Annual Conference of the PHM Society, volume 8, 2016, pp. 1–8.
- [80] W. Sun, S. Shao, R. Zhao, R. Yan, X. Zhang, X. Chen, A sparse auto-encoder-based deep neural network approach for induction motor faults classification, *Measurement* 89 (2016) 171–178.
- [81] W. Suwansin, P. Phasukkit, Deep learning-based acoustic emission scheme for nondestructive localization of cracks in train rails under a load, *Sensors* 21 (2021) 272.
- [82] C. Muir, B. Swaminathan, A. Almansour, K. Severer, C. Smith, M. Presby, J. Kiser, T. Pollock, S. Daly, Damage mechanism identification in composites via machine learning and acoustic emission, *npj Computational Materials* 7 (2021) 1–15.
- [83] P. Kane, A. Andhare, Application of psychoacoustics for gear fault diagnosis using artificial neural network, *Journal of Low Frequency Noise, Vibration and Active Control* 35 (2016) 207–220.
- [84] G. Abbiati, S. Marelli, C. Ligeikis, R. Christenson, B. Stojadinović, Training of a classifier for structural component failure based on hybrid simulation and kriging, *Journal of Engineering Mechanics* 148 (2022) 04021137.
- [85] T. Ritto, F. Rochinha, Digital twin, physics-based model, and machine learning applied to damage detection in structures, *Mechanical Systems and Signal Processing* 155 (2021) 107614.
- [86] X.-S. Si, W. Wang, C.-H. Hu, D.-H. Zhou, Remaining useful life estimation—a review on the statistical data driven approaches, *European journal of operational research* 213 (2011) 1–14.
- [87] A. K. Jardine, D. Lin, D. Banjevic, A review on machinery diagnostics and prognostics implementing condition-based maintenance, *Mechanical systems and signal processing* 20 (2006) 1483–1510.
- [88] A. Muneer, S. M. Taib, S. Naseer, R. F. Ali, I. A. Aziz, Data-driven deep learning-based attention mechanism for remaining useful life prediction: Case study application to turbofan engine analysis, *Electronics* 10 (2021) 2453.
- [89] B. Zhao, Q. Yuan, A novel deep learning scheme for multi-condition remaining useful life prediction of rolling element bearings, *Journal of Manufacturing Systems* 61 (2021) 450–460.
- [90] A. S. Yoon, T. Lee, Y. Lim, D. Jung, P. Kang, D. Kim, K. Park, Y. Choi, Semi-supervised learning with deep generative models for asset failure prediction, *arXiv preprint arXiv:1709.00845* (2017).
- [91] Y. Zhu, J. Wu, J. Wu, S. Liu, Dimensionality reduce-based for remaining useful life prediction of machining tools with multisensor fusion, *Reliability Engineering & System Safety* 218 (2022) 108179.
- [92] M. Farid, Data-driven method for real-time prediction and uncertainty quantification of fatigue failure under stochastic loading using artificial neural networks and gaussian process regression, *International Journal of Fatigue* 155 (2022) 106415.
- [93] L. Liao, W. Jin, R. Pavel, Enhanced restricted boltzmann machine with prognosability regularization for prognostics and health assessment, *IEEE Transactions on Industrial Electronics* 63 (2016) 7076–7083. doi:10.1109/TIE.2016.2586442.
- [94] W. Booyse, D. N. Wilke, S. Heyns, Deep digital twins for detection, diagnostics and prognostics, *Mechanical Systems and Signal Processing* 140 (2020) 106612.
- [95] P. M. Karve, Y. Guo, B. Kapusuzoglu, S. Mahadevan, M. A. Haile, Digital twin approach for damage-tolerant mission planning under uncertainty, *Engineering Fracture Mechanics* 225 (2020) 106766.
- [96] M. G. Kapteyn, D. J. Knezevic, K. Willcox, Toward predictive digital twins via component-based reduced-order models and interpretable machine learning, in: AIAA Scitech 2020 Forum, 2020, p. 0418.
- [97] M. Stender, M. Tiedemann, D. Spieler, D. Schoepflin, N. Hoffmann, S. Oberst, Deep learning for brake squeal: Brake noise detection, characterization and prediction, *Mechanical Systems and Signal Processing* 149 (2021) 107181.
- [98] V. Vidya, M. M. Dharmana, Model reference based intelligent control of an active suspension system for vehicles, in: 2017 International Conference on Circuit, Power and Computing Technologies (ICCPCT), IEEE, 2017, pp. 1–5.
- [99] J. Xu, J. Fei, Neural network predictive control of vehicle suspension, in: The 2nd International Conference on Information Science and Engineering, IEEE, 2010, pp. 1319–1322.
- [100] M. Jamil, M. N. Khan, S. J. Rind, Q. Awais, M. Uzair, Neural network predictive control of vibrations in tall structure: An experimental controlled vision, *Computers & Electrical Engineering* 89 (2021) 106940.
- [101] A. Saito, T. Kuno, Data-driven experimental modal analysis by dynamic mode decomposition, *Journal of Sound and Vibration* 481 (2020) 115434.
- [102] R. H. Cabell, C. Fuller, A principal component algorithm for feedforward active noise and vibration control, *Journal of Sound and Vibration* 227 (1999) 159–181.
- [103] B. Moore, Principal component analysis in linear systems: Controllability, observability, and model reduction, *IEEE transactions on automatic control* 26 (1981) 17–32.
- [104] R. Cabell, D. Palumbo, J. Viperman, A principal component feedforward algorithm for active noise control: flight test results, *IEEE Transactions on Control Systems Technology* 9 (2001) 76–83. doi:10.1109/87.896748.
- [105] Y. Hao, Comprehensive Robust Analysis for Controller Tuning in Principal Component Active Control Systems, Ph.D. thesis, University OF Leicester, 2020.
- [106] A. Al-Dmour, K. Mohammad, Active control of flexible structures using principal component analysis in the time domain, *Journal of sound and vibration* 253 (2002) 545–569.
- [107] M. Papadopoulos, E. Garcia, Sensor placement methodologies for dynamic testing, *AIAA journal* 36 (1998) 256–263.
- [108] Y. Liu, Y. Li, D. Wang, S. Zhang, Model updating of complex structures using the combination of component mode synthesis and kriging predictor, *The Scientific World Journal* 2014 (2014).
- [109] T. Simpson, N. Dervilis, E. Chatzi, Machine learning approach to model order reduction of nonlinear systems via autoencoder and lstm networks, *Journal of Engineering Mechanics* 147 (2021) 04021061. doi:10.1061/(asce)em.1943-7889.0001971.
- [110] S. Fresca, G. Gobat, P. Fedeli, A. Frangi, A. Manzoni, Deep learning-based reduced order models for the real-time simulation of the nonlinear dynamics of microstructures, *arXiv preprint arXiv:2111.12511* (2021).
- [111] L. Lu, K.-L. Yin, R. C. de Lamare, Z. Zheng, Y. Yu, X. Yang, B. Chen, A survey on active noise control in the past decade—part ii: Nonlinear systems, *Signal Processing* 181 (2021) 107929. doi:https://doi.org/10.1016/j.sigpro.2020.107929.
- [112] G. De Abreu, R. L. Teixeira, J. F. Ribeiro, A neural network-based direct inverse control for active control of vibrations of mechanical systems, in: Proceedings. Vol. 1. Sixth Brazilian Symposium on Neural Networks, IEEE, 2000, pp. 107–112.
- [113] W. C. Ariza-Zambrano, A. L. Serpa, Direct inverse control for active vibration suppression using artificial neural networks,

- Journal of Vibration and Control 27 (2021) 31–42.
- [114] A. C. Nerves, R. Krishnan, Active control of wind-induced vibrations in tall buildings using neural networks, in: Proceedings of IECON'94-20th Annual Conference of IEEE Industrial Electronics, volume 2, IEEE, 1994, pp. 1292–1297.
- [115] K. A. Bani-Hani, Vibration control of wind-induced response of tall buildings with an active tuned mass damper using neural networks, *Structural Control and Health Monitoring: The Official Journal of the International Association for Structural Control and Monitoring and of the European Association for the Control of Structures* 14 (2007) 83–108.
- [116] S. K. W. Park, Comparison of Neural Networks and Least Mean Squared Algorithms for Active Noise Canceling, Ph.D. thesis, Clemson University, 2018.
- [117] K. Zhang, G. Lyu, X. Luo, A deep recurrent neural network controller for nonlinear active noise control systems, in: 2020 IEEE 6th International Conference on Computer and Communications (ICCC), IEEE, 2020, pp. 2393–2396.
- [118] H. Zhang, D. Wang, Deep anc: A deep learning approach to active noise control, *Neural Networks* 141 (2021) 1–10. doi:https://doi.org/10.1016/j.neunet.2021.03.037.
- [119] C. T. Wangler, C. H. Hansen, Genetic algorithm adaptation of non-linear filter structures for active sound and vibration control, in: Proceedings of ICASSP'94. IEEE International Conference on Acoustics, Speech and Signal Processing, volume 3, IEEE, 1994, pp. III–505.
- [120] C.-Y. Chang, D.-R. Chen, Active noise cancellation without secondary path identification by using an adaptive genetic algorithm, *IEEE transactions on Instrumentation and Measurement* 59 (2010) 2315–2327.
- [121] M. A. Z. Raja, M. S. Aslam, N. I. Chaudhary, W. U. Khan, Bio-inspired heuristics hybrid with interior-point method for active noise control systems without identification of secondary path, *Frontiers of Information Technology & Electronic Engineering* 19 (2018) 246–259.
- [122] W. U. Khan, Z. Ye, N. I. Chaudhary, M. A. Z. Raja, Backtracking search integrated with sequential quadratic programming for nonlinear active noise control systems, *Applied Soft Computing* 73 (2018) 666–683.
- [123] M. A. Z. Raja, M. S. Aslam, N. I. Chaudhary, M. Nawaz, S. M. Shah, Design of hybrid nature-inspired heuristics with application to active noise control systems, *Neural Computing and Applications* 31 (2019) 2563–2591.
- [124] N. K. Rout, D. P. Das, G. Panda, Particle swarm optimization based nonlinear active noise control under saturation nonlinearity, *Applied Soft Computing* 41 (2016) 275–289.
- [125] N. V. George, G. Panda, A particle-swarm-optimization-based decentralized nonlinear active noise control system, *IEEE Transactions on Instrumentation and Measurement* 61 (2012) 3378–3386.
- [126] N. K. Rout, D. P. Das, G. Panda, Pso based adaptive narrow-band anc algorithm without the use of synchronization signal and secondary path estimate, *Mechanical Systems and Signal Processing* 114 (2019) 378–398.
- [127] M. Saad, Evolutionary optimization and real-time self-tuning active vibration control of a flexible beam system, Ph. D. thesis, Faculty of Mechanical Engineering, Universiti Teknologi Malaysia (2014).
- [128] H. Nobahari, S. A. Hosseini Kordkheili, S. S. Afshari, Hardware-in-the-loop optimization of an active vibration controller in a flexible beam structure using evolutionary algorithms, *Journal of Intelligent Material Systems and Structures* 25 (2014) 1211–1223.
- [129] A. G. A. Muthalif, K. A. M. Nor, A. N. Wahid, A. Ali, Optimization of piezoelectric sensor-actuator for plate vibration control using evolutionary computation: Modeling, simulation and experimentation, *IEEE Access* 9 (2021) 100725–100734. doi:10.1109/ACCESS.2021.3096972.
- [130] M. H. Awadalla, Spiking neural network and bull genetic algorithm for active vibration control, *International Journal of Intelligent Systems and Applications* 10 (2018) 17.
- [131] J. Katebi, M. Shoaie-parchin, M. Shariati, N. T. Trung, M. Khorami, Developed comparative analysis of metaheuristic optimization algorithms for optimal active control of structures, *Engineering with Computers* 36 (2020) 1539–1558.
- [132] Y.-Y. Lin, J.-Y. Chang, C.-T. Lin, A tsf-type-based self-evolving compensatory interval type-2 fuzzy neural network (tsfint2fnn) and its applications, *IEEE Transactions on Industrial Electronics* 61 (2013) 447–459.
- [133] Q.-Z. Zhang, W.-S. Gan, Y.-I. Zhou, Adaptive recurrent fuzzy neural networks for active noise control, *Journal of Sound and Vibration* 296 (2006) 935–948.
- [134] Q.-Z. Zhang, W.-S. Gan, Active noise control using a simplified fuzzy neural network, *Journal of sound and vibration* 1 (2004) 437–449.
- [135] N. Azadi, A. Ohadi, Filtered gradient active fuzzy neural network noise control in an enclosure backed by a clamped plate, *International Journal of Adaptive Control and Signal Processing* 26 (2012) 451–468.
- [136] S. D. Nguyen, Q. H. Nguyen, S.-B. Choi, Hybrid clustering based fuzzy structure for vibration control—part 1: A novel algorithm for building neuro-fuzzy system, *Mechanical Systems and Signal Processing* 50 (2015) 510–525.
- [137] D. Singh, Passenger body vibration control in active quarter car model using anfis based super twisting sliding mode controller, *Simulation Modelling Practice and Theory* 89 (2018) 100–118.
- [138] L. Buşoniu, T. de Bruin, D. Tolić, J. Kober, I. Palunko, Reinforcement learning for control: Performance, stability, and deep approximators, *Annual Reviews in Control* 46 (2018) 8–28. doi:https://doi.org/10.1016/j.arcontrol.2018.09.005.
- [139] K. Latifi, A. Kopitca, Q. Zhou, Model-free control for dynamic-field acoustic manipulation using reinforcement learning, *IEEE Access* 8 (2020) 20597–20606.
- [140] B. Raesy, S. G. Haghighi, Active noise controller with reinforcement learning, in: The 16th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP 2012), IEEE, 2012, pp. 074–079.
- [141] Z.-c. Qiu, G.-h. Chen, X.-m. Zhang, Reinforcement learning vibration control for a flexible hinged plate, *Aerospace Science and Technology* 118 (2021) 107056.
- [142] Z. Tao, D. Yian, H. Fan, Z. Xiangqi, W. Yaoyao, L. Tianlin, P. Bolchover, Y. Tao, Z. Guishui, C. Rongbing, et al., Reducing vibration of a rotating machine with deep reinforcement learning, in: 2020 IEEE International Conference on Mechatronics and Automation (ICMA), IEEE, 2020, pp. 932–937.
- [143] R. Gulde, M. Tuscher, A. Csizsar, O. Riedel, A. Verl, Reinforcement learning approach to vibration compensation for dynamic feed drive systems, in: 2019 Second International Conference on Artificial Intelligence for Industries (AI4I), IEEE, 2019, pp. 26–29.
- [144] S. S. Eshkevari, S. S. Eshkevari, D. Sen, S. N. Pakzad, RL-controller: a reinforcement learning framework for active structural control, arXiv preprint arXiv:2103.07616 (2021).
- [145] H. Gao, W. He, Y. Zhang, C. Sun, Vibration control based on reinforcement learning for a flexible building-like structure system with active mass damper against disturbance effects, in: 2020 59th IEEE Conference on Decision and Control (CDC), IEEE, 2020, pp. 2380–2385.
- [146] A. Cicirello, B. R. Mace, M. J. Kingan, Y. Yang, Sensitivity analysis of generalised eigenproblems and application to wave

- and finite element models, *Journal of Sound and Vibration* 478 (2020) 115345.
- [147] D. E. Tsokaktsidis, T. Von Wysocki, F. Gauterin, S. Marburg, Artificial Neural Network predicts noise transfer as a function of excitation and geometry, *Universitätsbibliothek der RWTH Aachen*, 2019.
- [148] D. Birky, J. Ladd, I. Guardiola, A. Young, Predicting the dynamic response of a structure using an artificial neural network, *Journal of Low Frequency Noise, Vibration and Active Control* (2021) 14613484211038408.
- [149] R. Zhang, Y. Liu, H. Sun, Physics-guided convolutional neural network (phycnn) for data-driven seismic response modeling, *Engineering Structures* 215 (2020) 110704.
- [150] A. Casaburo, G. Petrone, V. Meruane, F. Franco, S. D. Rosa, Evaluation of plates in similitude by experimental tests and artificial neural networks, *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* (2021) 095444062211002551.
- [151] W. Chai, A. Saidi, A. Zine, C. Droz, W. You, M. Ichchou, Comparison of uncertainty quantification process using statistical and data mining algorithms, *Structural and Multidisciplinary Optimization* 61 (2020) 587–598.
- [152] G. Abbiati, S. Marelli, N. Tsokanas, B. Sudret, B. Stojadinović, A global sensitivity analysis framework for hybrid simulation, *Mechanical Systems and Signal Processing* 146 (2021) 106997.
- [153] A. Nobari, H. Ouyang, P. Bannister, Uncertainty quantification of squeal instability via surrogate modelling, *Mechanical Systems and Signal Processing* 60 (2015) 887–908.
- [154] J. E. Hurtado, D. A. Alvarez, Neural-network-based reliability analysis: a comparative study, *Computer methods in applied mechanics and engineering* 191 (2001) 113–132.
- [155] L. Wang, Z. Chen, G. Yang, An interval uncertainty analysis method for structural response bounds using feedforward neural network differentiation, *Applied Mathematical Modelling* 82 (2020) 449–468.
- [156] C. Soize, *Uncertainty quantification*, Springer, 2017.
- [157] J.-Q. Liu, Y.-W. Feng, X.-F. Xue, C. Lu, Intelligent extremum surrogate modeling framework for dynamic probabilistic analysis of complex mechanism, *Mathematical Problems in Engineering* 2021 (2021).
- [158] C. Lu, C.-W. Fei, Y.-W. Feng, Y.-J. Zhao, X.-W. Dong, Y.-S. Choy, Probabilistic analyses of structural dynamic response with modified kriging-based moving extremum framework, *Engineering Failure Analysis* 125 (2021) 105398.
- [159] C. Lu, Y.-W. Feng, R. P. Liem, C.-W. Fei, Improved kriging with extremum response surface method for structural dynamic reliability and sensitivity analyses, *Aerospace Science and Technology* 76 (2018) 164–175.
- [160] Q. Guo, Y. Liu, Y. Zhao, B. Li, Q. Yao, Improved resonance reliability and global sensitivity analysis of multi-span pipes conveying fluid based on active learning kriging model, *International Journal of Pressure Vessels and Piping* 170 (2019) 92–101.
- [161] Q. Guo, Y. Liu, B. Chen, Q. Yao, A variable and mode sensitivity analysis method for structural system using a novel active learning kriging model, *Reliability Engineering & System Safety* 206 (2021) 107285. doi:<https://doi.org/10.1016/j.res.2020.107285>.
- [162] W. You, A. Saidi, A.-m. Zine, M. Ichchou, Mechanical reliability assessment by ensemble learning, *Vehicles* 2 (2020) 126–141.
- [163] K. Craig, N. Stander, D. Dooge, S. Varadappa, Mdo of automotive vehicle for crashworthiness and nvh using response surface methods, in: *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 2002, p. 5607.
- [164] M. Kiani, A. R. Yildiz, A comparative study of non-traditional methods for vehicle crashworthiness and nvh optimization, *Archives of Computational Methods in Engineering* 23 (2016) 723–734.
- [165] I. Ibrahim, R. Silva, M. Mohammadi, V. Ghorbanian, D. A. Lowther, Surrogate-based acoustic noise prediction of electric motors, *IEEE Transactions on Magnetics* 56 (2020) 1–4.
- [166] J. Zhang, W. Chu, J. Zhang, Y. Lv, Vibroacoustic optimization study for the volute casing of a centrifugal fan, *Applied Sciences* 9 (2019) 859.
- [167] K.-J. Cha, C.-U. Chin, J.-S. Ryu, J.-E. Oh, The optimal design for low noise intake system using kriging method with robust design, *JSME International Journal Series C Mechanical Systems, Machine Elements and Manufacturing* 47 (2004) 873–881.
- [168] A. Casaburo, D. Magliacano, G. Petrone, F. Franco, S. De Rosa, Optimizing the acoustic properties of a metamaterial using machine learning techniques, in: *INTER-NOISE and NOISE-CON Congress and Conference Proceedings*, volume 263, Institute of Noise Control Engineering, 2021, pp. 3044–3055.
- [169] A. Bacigalupo, G. Gnecco, M. Lepidi, L. Gambarotta, Machine-learning techniques for the optimal design of acoustic metamaterials, *Journal of Optimization Theory and Applications* 187 (2020) 630–653.
- [170] I. Park, D. Papadimitriou, Efficient surrogate-based nvh optimization of a full vehicle using frf based substructuring, *SAE International Journal of Advances and Current Practices in Mobility* 2 (2020) 1429–1442.
- [171] M. Li, W. Zhou, J. Liu, X. Zhang, F. Pan, H. Yang, M. Li, D. Luo, Vehicle interior noise prediction based on elman neural network, *Applied Sciences* 11 (2021). doi:[10.3390/app11178029](https://doi.org/10.3390/app11178029).
- [172] D. E. Tsokaktsidis, C. Nau, S. Marburg, Time Domain Full Vehicle Interior Noise Calculation from Component Level Data by Machine Learning, Technical Report, SAE Technical Paper, 2020.
- [173] T. von Wysocki, F. Rieger, D. E. Tsokaktsidis, F. Gauterin, Generating component designs for an improved nvh performance by using an artificial neural network as an optimization metamodel, *Designs* 5 (2021). doi:[10.3390/designs5020036](https://doi.org/10.3390/designs5020036).
- [174] T. von Wysocki, M. Leupolz, F. Gauterin, Metamodels resulting from two different geometry morphing approaches are suitable to direct the modification of structure-born noise transfer in the digital design phase, *Applied System Innovation* 3 (2020) 47.
- [175] P. Mohanasundaram, F. Gillot, K. Shimoyama, S. Besset, Shape optimization of a disc-pad system under squeal noise criteria, *SN Applied Sciences* 2 (2020) 1–15.
- [176] D. Du, E. He, F. Li, D. Huang, Using the hierarchical kriging model to optimize the structural dynamics of rocket engines, *Aerospace Science and Technology* 107 (2020) 106248. doi:<https://doi.org/10.1016/j.ast.2020.106248>.
- [177] A. Bacigalupo, G. Gnecco, M. Lepidi, L. Gambarotta, Computational design of innovative mechanical metafilters via adaptive surrogate-based optimization, *Computer Methods in Applied Mechanics and Engineering* 375 (2021) 113623.
- [178] M. Moustapha, B. Sudret, Surrogate-assisted reliability-based design optimization: a survey and a unified modular framework, *Structural and Multidisciplinary Optimization* (2019) 1–20.
- [179] C.-W. Fei, W.-Z. Tang, G.-C. Bai, Novel method and model for dynamic reliability optimal design of turbine blade deformation, *Aerospace Science and Technology* 39 (2014) 588–595.

- [180] C.-Y. Zhang, Z.-S. Yuan, Z. Wang, C.-W. Fei, C. Lu, Probabilistic fatigue/creep optimization of turbine bladed disk with fuzzy multi-extremum response surface method, *Materials* 12 (2019) 3367.
- [181] F. F. Nascentes, R. H. Lopez, J. E. S. Cursi, R. Sampaio, L. F. Miguel, An efficient global optimization approach for reliability maximization of friction-tuned mass damper-controlled structures, *Shock and Vibration* 2018 (2018).
- [182] P. Domingos, A few useful things to know about machine learning, *Communications of the ACM* 55 (2012) 78–87.
- [183] I. Goodfellow, Y. Bengio, A. Courville, *Deep learning*, MIT press, 2016.
- [184] J. Bergstra, R. Bardenet, Y. Bengio, B. Kégl, Algorithms for hyper-parameter optimization, *Advances in neural information processing systems* 24 (2011).
- [185] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization., *Journal of machine learning research* 13 (2012).
- [186] G. A. Lujan-Moreno, P. R. Howard, O. G. Rojas, D. C. Montgomery, Design of experiments and response surface methodology to tune machine learning hyperparameters, with a random forest case-study, *Expert Systems with Applications* 109 (2018) 195–205.
- [187] M. Feurer, F. Hutter, Hyperparameter optimization, in: *Automated machine learning*, Springer, Cham, 2019, pp. 3–33.
- [188] J. Bergstra, B. Komer, C. Eliasmith, D. Yamins, D. D. Cox, Hyperopt: a python library for model selection and hyperparameter optimization, *Computational Science & Discovery* 8 (2015) 014008.
- [189] C. M. Bishop, *Pattern recognition*, *Machine learning* 128 (2006).
- [190] J. Friedman, T. Hastie, R. Tibshirani, et al., *The elements of statistical learning*, 10, Springer series in statistics New York, 2001.
- [191] G. Cybenko, Approximation by superpositions of a sigmoidal function, *Mathematics of control, signals and systems* 2 (1989) 303–314.
- [192] M. A. Nielsen, *Neural networks and deep learning*, volume 25, Determination press San Francisco, CA, 2015.
- [193] R. Reed, R. J. MarksII, *Neural smithing: supervised learning in feedforward artificial neural networks*, Mit Press, 1999.
- [194] F. Chollet, *Deep learning with Python*, Simon and Schuster, 2021.
- [195] Y. LeCun, Y. Bengio, G. Hinton, *Deep learning*, *nature* 521 (2015) 436–444.
- [196] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, J. M. Siskind, Automatic differentiation in machine learning: a survey, *Journal of machine learning research* 18 (2018).
- [197] E.-G. Talbi, Automated design of deep neural networks: A survey and unified taxonomy, *ACM Computing Surveys (CSUR)* 54 (2021) 1–37.
- [198] Y. A. LeCun, L. Bottou, G. B. Orr, K.-R. Müller, Efficient backprop, in: *Neural networks: Tricks of the trade*, Springer, 2012, pp. 9–48.
- [199] S. Sivanandam, S. Deepa, *Introduction to neural networks using Matlab 6.0*, Tata McGraw-Hill Education, 2006.
- [200] E. Stevens, L. Antiga, T. Viehmann, *Deep learning with PyTorch*, Manning Publications, 2020.
- [201] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, M. Hasan, B. C. Van Essen, A. A. Awwal, V. K. Asari, A state-of-the-art survey on deep learning theory and architectures, *Electronics* 8 (2019) 292.
- [202] J. Schmidhuber, *Deep learning in neural networks: An overview*, *Neural networks* 61 (2015) 85–117.
- [203] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, B. B. Gupta, X. Chen, X. Wang, A survey of deep active learning, *ACM Computing Surveys (CSUR)* 54 (2021) 1–40.
- [204] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, Q. He, A comprehensive survey on transfer learning, *Proceedings of the IEEE* 109 (2020) 43–76.
- [205] S. C. Hoi, D. Sahoo, J. Lu, P. Zhao, Online learning: A comprehensive survey, *Neurocomputing* 459 (2021) 249–289.
- [206] S. Bachstein, *Uncertainty quantification in deep learning*, Master's thesis, ULM University, 2019.
- [207] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [208] Z. J. Wang, R. Turko, O. Shaikh, H. Park, N. Das, F. Hohman, M. Kahng, D. H. Chau, Cnn 101: Interactive visual learning for convolutional neural networks, in: *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020, pp. 1–7.
- [209] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, et al., Recent advances in convolutional neural networks, *Pattern Recognition* 77 (2018) 354–377.
- [210] H. Hewamalage, C. Bergmeir, K. Bandara, Recurrent neural networks for time series forecasting: Current status and future directions, *International Journal of Forecasting* 37 (2021) 388–427. doi:<https://doi.org/10.1016/j.ijforecast.2020.06.008>.
- [211] R. Zhang, Y. Liu, H. Sun, Physics-informed multi-lstm networks for metamodeling of nonlinear structures, *Computer Methods in Applied Mechanics and Engineering* 369 (2020) 113226.
- [212] Y. Zhang, P. Hutchinson, N. A. Lieven, J. Nunez-Yanez, Remaining useful life estimation using long short-term memory neural networks and deep fusion, *IEEE Access* 8 (2020) 19033–19045.
- [213] C. Olah, *Understanding lstm networks*, [shorturl.at/cgzDF](https://arxiv.org/abs/1901.10147), 2015. Accessed: 2022-01-17.
- [214] A. Hochlehnert, A. Terenin, S. Sæmundsson, M. Deisenroth, Learning contact dynamics using physically structured neural networks, in: *International Conference on Artificial Intelligence and Statistics*, PMLR, 2021, pp. 2152–2160.
- [215] A. Graves, *Supervised Sequence Labelling with Recurrent Neural Networks*, *Studies in computational intelligence*, Springer, Berlin, 2012. doi:10.1007/978-3-642-24797-2.
- [216] H. Salehinejad, S. Sankar, J. Barfett, E. Colak, S. Valaee, Recent advances in recurrent neural networks, *arXiv preprint arXiv:1801.01078* (2017).
- [217] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, Y. Bengio, Show, attend and tell: Neural image caption generation with visual attention, in: *International conference on machine learning*, PMLR, 2015, pp. 2048–2057.
- [218] T. Daniel, F. Casenave, N. Akkari, D. Ryckelynck, Model order reduction assisted by deep neural networks (rom-net), *Advanced Modeling and Simulation in Engineering Sciences* 7 (2020) 1–27.
- [219] T. Lassila, A. Manzoni, A. Quarteroni, G. Rozza, Model order reduction in fluid dynamics: challenges and perspectives, *Reduced Order Methods for modeling and computational reduction* (2014) 235–273.
- [220] G. E. Hinton, R. R. Salakhutdinov, Reducing the dimensionality of data with neural networks, *science* 313 (2006) 504–507.
- [221] M. Sakurada, T. Yairi, Anomaly detection using autoencoders with nonlinear dimensionality reduction, in: *Proceedings of the MLSDA 2014 2nd workshop on machine learning for sensory data analysis*, 2014, pp. 4–11.
- [222] W.-Y. Loh, *Classification and regression trees*, *Wiley interdisciplinary reviews: data mining and knowledge discovery* 1 (2011) 14–23.

- [223] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, *Journal of Machine Learning Research* 12 (2011) 2825–2830.
- [224] G. Louppe, Understanding random forests: From theory to practice, *arXiv preprint arXiv:1407.7502* (2014).
- [225] P. Roßbach, Neural networks vs. random forests—does it always have to be deep learning, Germany: Frankfurt School of Finance and Management (2018).
- [226] W. Chai, Global sensitivity analysis on vibro-acoustic composite materials with parametric dependency, Ph.D. thesis, Lyon, 2018.
- [227] A. Natekin, A. Knoll, Gradient boosting machines, a tutorial, *Frontiers in neurorobotics* 7 (2013) 21.
- [228] T. Chen, C. Guestrin, XGBoost: A scalable tree boosting system, in: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16*, ACM, New York, NY, USA, 2016, pp. 785–794. URL: <http://doi.acm.org/10.1145/2939672.2939785>. doi:10.1145/2939672.2939785.
- [229] Y. Freund, R. E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, *Journal of Computer and System Sciences* 55 (1997) 119–139. doi:<https://doi.org/10.1006/jcss.1997.1504>.
- [230] XGBoost-developers, Introduction to boosted trees, [shorturl.at/iyBJK](https://shorturl.at/iyBJK), 2021. Accessed:2021-12-23.
- [231] T. Beckers, An introduction to gaussian process models, *arXiv preprint arXiv:2102.05497* (2021).
- [232] C. E. Rasmussen, Gaussian processes in machine learning, in: *Summer school on machine learning*, Springer, 2003, pp. 63–71.
- [233] M. A. Bouhlel, J. T. Hwang, N. Bartoli, R. Lafage, J. Morlier, J. R. R. A. Martins, A python surrogate modeling framework with derivatives, *Advances in Engineering Software* (2019) 102662. doi:<https://doi.org/10.1016/j.advengsoft.2019.03.005>.
- [234] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, et al., A review of uncertainty quantification in deep learning: Techniques, applications and challenges, *Information Fusion* (2021).
- [235] P. I. Frazier, Bayesian optimization, in: *Recent Advances in Optimization and Modeling of Contemporary Problems, INFORMS*, 2018, pp. 255–278.
- [236] D. R. Jones, M. Schonlau, W. J. Welch, Efficient global optimization of expensive black-box functions, *Journal of Global optimization* 13 (1998) 455–492.
- [237] R. Xu, D. Wunsch, Survey of clustering algorithms, *IEEE Transactions on neural networks* 16 (2005) 645–678.
- [238] Scikit-learn, Clustering [documentation], [shorturl.at/igu38](https://shorturl.at/igu38), 2021. Accessed: 2021-12-23.
- [239] R. S. Sutton, A. G. Barto, Reinforcement learning: An introduction, MIT press, 2018.
- [240] Y. Li, Deep reinforcement learning, 2018. *arXiv:1810.06339*.
- [241] C. J. C. H. Watkins, Learning from delayed rewards, Ph.D. thesis, King's College, Cambridge United Kingdom, 1989.
- [242] J. Clifton, E. Laber, Q-learning: Theory and applications, *Annual Review of Statistics and Its Application* 7 (2020) 279–301.
- [243] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra, Continuous control with deep reinforcement learning, *arXiv preprint arXiv:1509.02971* (2015).
- [244] Y. Tassa, Y. Doron, A. Muldal, T. Erez, Y. Li, D. d. L. Casas, D. Budden, A. Abdolmaleki, J. Merel, A. Lefrancq, et al., Deepmind control suite, *arXiv preprint arXiv:1801.00690* (2018).
- [245] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al., Human-level control through deep reinforcement learning, *nature* 518 (2015) 529–533.
- [246] F.-G. Yuan, S. A. Zargar, Q. Chen, S. Wang, Machine learning for structural health monitoring: challenges and opportunities, in: *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2020*, volume 11379, International Society for Optics and Photonics, 2020, p. 1137903.
- [247] Y. Lei, B. Yang, X. Jiang, F. Jia, N. Li, A. K. Nandi, Applications of machine learning to machine fault diagnosis: A review and roadmap, *Mechanical Systems and Signal Processing* 138 (2020) 106587.
- [248] S. Shao, W. Sun, P. Wang, R. X. Gao, R. Yan, Learning features from vibration signals for induction motor fault diagnosis, in: *2016 International Symposium on Flexible Automation (ISFA)*, IEEE, 2016, pp. 71–76.
- [249] F. Zhou, Y. Gao, C. Wen, A novel multimode fault classification method based on deep learning, *Journal of Control Science and Engineering* 2017 (2017).
- [250] Z. Huijie, R. Ting, W. Xinqing, Z. You, F. Husheng, Fault diagnosis of hydraulic pump based on stacked autoencoders, in: *2015 12th IEEE International Conference on Electronic Measurement & Instruments (ICEMI)*, volume 1, IEEE, 2015, pp. 58–62.
- [251] Z. Chen, S. Deng, X. Chen, C. Li, R.-V. Sanchez, H. Qin, Deep neural networks-based rolling bearing fault diagnosis, *Microelectronics Reliability* 75 (2017) 327–333.
- [252] M. Varanis, R. Pederiva, Statements on wavelet packet energy-entropy signatures and filter influence in fault diagnosis of induction motor in non-stationary operations, *Journal of the Brazilian Society of Mechanical Sciences and Engineering* 40 (2018) 1–10.
- [253] X. Ding, Q. He, Energy-fluctuated multiscale feature learning with deep convnet for intelligent spindle bearing fault diagnosis, *IEEE Transactions on Instrumentation and Measurement* 66 (2017) 1926–1935.
- [254] J. Ma, J. Wu, X. Wang, Fault diagnosis method based on wavelet packet-energy entropy and fuzzy kernel extreme learning machine, *Advances in Mechanical Engineering* 10 (2018) 1687814017751446.
- [255] M. Sohaib, C.-H. Kim, J.-M. Kim, A hybrid feature model and deep-learning-based bearing fault diagnosis, *Sensors* 17 (2017) 2876.
- [256] L. Guo, H. Gao, H. Huang, X. He, S. Li, Multifeatures fusion and nonlinear dimension reduction for intelligent bearing condition monitoring, *Shock and Vibration* 2016 (2016).
- [257] N. K. Verma, V. K. Gupta, M. Sharma, R. K. Sevakula, Intelligent condition based monitoring of rotating machines using sparse auto-encoders, in: *2013 IEEE Conference on prognostics and health management (PHM)*, IEEE, 2013, pp. 1–7.
- [258] H. Oh, J. H. Jung, B. C. Jeon, B. D. Youn, Scalable and unsupervised feature engineering using vibration-imaging and deep learning for rotor system diagnosis, *IEEE Transactions on Industrial Electronics* 65 (2017) 3539–3549.
- [259] Y. Zhang, Z. Zhao, H.-j. Xu, C. He, H. Peng, Z. Gao, Z.-y. Xu, Vibration pattern recognition using a compressed histogram of oriented gradients for snoring source analysis, *Bio-Medical Materials and Engineering* 31 (2020) 143–155.
- [260] K. Worden, G. Manson, The application of machine learning to structural health monitoring, *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering*

- Sciences 365 (2007) 515–537.
- [261] C. Lu, Z.-Y. Wang, W.-L. Qin, J. Ma, Fault diagnosis of rotary machinery components using a stacked denoising autoencoder-based health state identification, *Signal Processing* 130 (2017) 377–388.
- [262] MathWorks, Wind turbine high-speed bearing prognosis (r2012b), [shorturl.at/lnCNW](https://shorturl.at/lnCNW), 2021. Accessed: 2021-11-23.
- [263] MathWorks, Similarity-based remaining useful life estimation (r2012b), [shorturl.at/ivQZ4](https://shorturl.at/ivQZ4), 2021. Accessed: 2021-11-23.
- [264] L. Meirovitch, *Dynamics and control of structures*, John Wiley & Sons, 1991.
- [265] C. C. Fuller, S. J. Elliott, P. A. Nelson, *Active control of vibration*, Academic press, 1996.
- [266] W. Gawronski, *Advanced structural dynamics and active control of structures*, Springer Science & Business Media, 2004.
- [267] R. Alkhatib, M. Golnaraghi, *Active structural vibration control: a review*, *Shock and Vibration Digest* 35 (2003) 367.
- [268] C. Fuller, A. Von Flotow, *Active control of sound and vibration*, *IEEE Control Systems Magazine* 15 (1995) 9–19.
- [269] S. Elliott, P. Nelson, *The active control of sound*, *Electronics & communication engineering journal* 2 (1990) 127–136.
- [270] J. Cheer, *Active sound control in the automotive interior*, *Future Interior Concepts* (2021) 53–69.
- [271] C. Hansen, S. Snyder, X. Qiu, L. Brooks, D. Moreau, *Active Control of Noise and Vibration*, CRC Press, 2012.
- [272] S. L. Brunton, J. N. Kutz, *Data-driven science and engineering: Machine learning, dynamical systems, and control*, Cambridge University Press, 2019.
- [273] A. S. Umar, M. D. Alhassan, K. Aminu, S. G. Ahmad, *Modelling and Control of Dynamical Systems Using Neural Network – A Review*, *International Journal of Soft Computing and Engineering (IJSCCE)* 5 (2015).
- [274] W. T. Miller, P. J. Werbos, R. S. Sutton, *Neural networks for control*, MIT press, 1995.
- [275] K. J. Hunt, D. Sbarbaro, R. Żbikowski, P. J. Gawthrop, *Neural networks for control systems—a survey*, *Automatica* 28 (1992) 1083–1112.
- [276] D. Soloway, P. J. Haley, *Neural generalized predictive control*, in: *Proceedings of the 1996 IEEE international symposium on intelligent control*, IEEE, 1996, pp. 277–282.
- [277] K. S. Narendra, S. Mukhopadhyay, *Adaptive control using neural networks and approximate models*, *IEEE Transactions on neural networks* 8 (1997) 475–485.
- [278] S. N. Kumpati, P. Kannan, et al., *Identification and control of dynamical systems using neural networks*, *IEEE Transactions on neural networks* 1 (1990) 4–27.
- [279] S. Brunton, *Data-driven control with machine learning* [youtube playlist], [shorturl.at/kmBV4](https://shorturl.at/kmBV4), 2020. Accessed: 2022-01-06.
- [280] D. Kim, S.-H. Kim, T. Kim, B. B. Kang, M. Lee, W. Park, S. Ku, D. Kim, J. Kwon, H. Lee, et al., *Review of machine learning methods in soft robotics*, *Plos one* 16 (2021) e0246102.
- [281] L. Ljung, C. Andersson, K. Tiels, T. B. Schön, *Deep learning and system identification*, *IFAC-PapersOnLine* 53 (2020) 1175–1181.
- [282] T. Duriez, S. L. Brunton, B. R. Noack, *Machine learning control-taming nonlinear dynamics and turbulence*, Springer, 2017.
- [283] A. Chiuso, G. Pillonetto, *System identification: A machine learning perspective*, *Annual Review of Control, Robotics, and Autonomous Systems* 2 (2019) 281–304.
- [284] G. Pillonetto, F. Dinuzzo, T. Chen, G. De Nicolao, L. Ljung, *Kernel methods in system identification, machine learning and function estimation: A survey*, *Automatica* 50 (2014) 657–682.
- [285] W. Liu, P. P. Pokharel, J. C. Principe, *The kernel least-mean-square algorithm*, *IEEE Transactions on Signal Processing* 56 (2008) 543–554. doi:10.1109/TSP.2007.907881.
- [286] W. Liu, J. C. Principe, *Kernel affine projection algorithms*, *EURASIP Journal on Advances in Signal Processing* 2008 (2008) 1–12.
- [287] H. T. Siegelmann, B. G. Horne, C. L. Giles, *Computational capabilities of recurrent narx neural networks*, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 27 (1997) 208–215.
- [288] M. Schüssler, T. Munker, O. Nelles, *Local model networks for the identification of nonlinear state space models*, in: *2019 IEEE 58th Conference on Decision and Control (CDC)*, IEEE, 2019, pp. 6437–6442.
- [289] E. Skomski, J. Drgoňa, A. Tuor, *Automating discovery of physics-informed neural state space models via learning and evolution*, in: *Learning for Dynamics and Control*, PMLR, 2021, pp. 980–991.
- [290] M. S. Alhajeri, Z. Wu, D. Rincon, F. Albalawi, P. D. Christofides, *Machine-learning-based state estimation and predictive control of nonlinear processes*, *Chemical Engineering Research and Design* 167 (2021) 268–280.
- [291] J. N. Kutz, S. L. Brunton, B. W. Brunton, J. L. Proctor, *Dynamic mode decomposition: data-driven modeling of complex systems*, SIAM, 2016.
- [292] O. Azencot, N. B. Erichson, V. Lin, M. Mahoney, *Forecasting sequential data using consistent koopman autoencoders*, in: *International Conference on Machine Learning*, PMLR, 2020, pp. 475–485.
- [293] S. L. Brunton, J. L. Proctor, J. N. Kutz, *Discovering governing equations from data by sparse identification of nonlinear dynamical systems*, *Proceedings of the national academy of sciences* 113 (2016) 3932–3937.
- [294] M. Didonna, M. Stender, A. Papangelo, F. Fontanela, M. Ciavarella, N. Hoffmann, *Reconstruction of governing equations from vibration measurements for geometrically nonlinear systems*, *Lubricants* 7 (2019) 64.
- [295] P. Baldi, K. Hornik, *Neural networks and principal component analysis: Learning from examples without local minima*, *Neural networks* 2 (1989) 53–58.
- [296] H. Li, D. Fernex, R. Semaan, J. Tan, M. Morzyński, B. R. Noack, *Cluster-based network model*, *Journal of Fluid Mechanics* 906 (2021).
- [297] F. L. Lewis, D. Vrabie, K. G. Vamvoudakis, *Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers*, *IEEE Control Systems Magazine* 32 (2012) 76–105.
- [298] J. Kober, J. A. Bagnell, J. Peters, *Reinforcement learning in robotics: A survey*, *The International Journal of Robotics Research* 32 (2013) 1238–1274.
- [299] Á. Bárkányi, T. Chován, S. Németh, J. Abonyi, *Modelling for digital twins—potential role of surrogate models*, *Processes* 9 (2021). doi:10.3390/pr9030476.
- [300] S. Marelli, P.-R. Wagner, C. Lataniotis, B. Sudret, *Stochastic spectral embedding*, *International Journal for Uncertainty Quantification* 11 (2021).
- [301] H. Liu, Y. S. Ong, J. Cai, *A survey of adaptive sampling for global metamodelling in support of simulation-based complex engineering design*, *Structural and Multidisciplinary Optimization* 57 (2018) 393–416. doi:10.1007/s00158-017-1739-8.
- [302] J. Willard, X. Jia, S. Xu, M. Steinbach, V. Kumar, *Integrating scientific knowledge with machine learning for engineering and environmental systems*, 2021. arXiv:2003.04919.

- [303] M. Böttcher, F. Leichsenring, A. Fuchs, W. Graf, M. Kaliske, Efficient utilization of surrogate models for uncertainty quantification, *PAMM* 20 (2021) e202000210.
- [304] B. Sudret, S. Marelli, J. Wiart, Surrogate models for uncertainty quantification: An overview, in: 2017 11th European conference on antennas and propagation (EUCAP), IEEE, 2017, pp. 793–797.
- [305] R. P. Dwight, J. de Baar, I. Azizli, A tutorial on adaptive surrogate modelling for global optimization, *Dimension* 8 (2012) 7.
- [306] F. Xiong, S. Chen, Y. Xiong, Dynamic system uncertainty propagation using polynomial chaos, *Chinese Journal of Aeronautics* 27 (2014) 1156–1170.
- [307] B. Sudret, Global sensitivity analysis using polynomial chaos expansions, *Reliability engineering & system safety* 93 (2008) 964–979.
- [308] H.-M. Gutmann, A radial basis function method for global optimization, *Journal of global optimization* 19 (2001) 201–227.
- [309] K. Chaiyotha, T. Krityakierne, A comparative study of infill sampling criteria for computationally expensive constrained optimization problems, *Symmetry* 12 (2020) 1631.
- [310] M. T. Emmerich, K. Yang, A. H. Deutz, Infill criteria for multiobjective bayesian optimization, in: *High-Performance Simulation-Based Optimization*, Springer, 2020, pp. 3–16.
- [311] S. Chakraborty, S. Adhikari, R. Ganguli, The role of surrogate models in the development of digital twins of dynamic systems, *Applied Mathematical Modelling* 90 (2021) 662–681.
- [312] P. Gardner, M. Dal Borgo, V. Ruffini, A. J. Hughes, Y. Zhu, D. J. Wagg, Towards the development of an operational digital twin, *Vibration* 3 (2020) 235–265.
- [313] M. Cao, N. F. Alkayem, L. Pan, D. Novák, Advanced methods in neural networks-based sensitivity analysis with their applications in civil engineering, *Artificial neural networks: models and applications*, Rijeka, Croatia, IntechOpen (2016) 335–353.
- [314] K. Cheng, Z. Lu, C. Ling, S. Zhou, Surrogate-assisted global sensitivity analysis: an overview, *Structural and Multidisciplinary Optimization* 61 (2020) 1187–1213.
- [315] J. Pizarroso, J. Portela, A. Muñoz, Neursens: Sensitivity analysis of neural networks, *arXiv preprint arXiv:2002.11423* (2020).
- [316] A. Tank, I. Covert, N. Foti, A. Shojaie, E. B. Fox, Neural granger causality, *IEEE Transactions on Pattern Analysis & Machine Intelligence* (2021) 1–1.
- [317] M. Böhle, F. Eitel, M. Weygandt, K. Ritter, Layer-wise relevance propagation for explaining deep neural network decisions in mri-based alzheimer’s disease classification, *Frontiers in aging neuroscience* 11 (2019) 194.
- [318] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, W. Samek, On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation, *PloS one* 10 (2015) e0130140.
- [319] B. Bhattacharyya, E. Jacquelin, D. Brizard, Uncertainty quantification of nonlinear stochastic dynamic problem using a kriging-narx surrogate model, in: *3rd International Conference on Uncertainty Quantification in Computational Sciences and Engineering*, ECCOMAS, 2019, pp. 13–p.
- [320] R. K. Tripathy, I. Bilionis, Deep uq: Learning deep neural network surrogate models for high dimensional uncertainty quantification, *Journal of computational physics* 375 (2018) 565–588.
- [321] X. Luo, A. Kareem, Deep convolutional neural networks for uncertainty propagation in random fields, *Computer-Aided Civil and Infrastructure Engineering* 34 (2019) 1043–1054.
- [322] A. Chaudhuri, R. Lam, K. Willcox, Multifidelity uncertainty propagation via adaptive surrogates in coupled multidisciplinary systems, *AIAA Journal* 56 (2018) 235–249.
- [323] R. Guo, T. Mi, L. Li, R. Luo, Research on aerodynamic performance and noise reduction of high-voltage fans on fuel cell vehicles, *Applied Acoustics* 186 (2022) 108454.
- [324] J. Berkeley, H. B. Moss, A. Artemev, S. Pascual-Diaz, U. Granta, H. Stojic, I. Couckuyt, J. Qing, L. Satrio, V. Picheny, Trieste, 2021. URL: <https://github.com/secondmind-labs/trieste>.
- [325] M. Balandat, B. Karrer, D. R. Jiang, S. Daulton, B. Letham, A. G. Wilson, E. Bakshy, Botorch: A framework for efficient monte-carlo bayesian optimization., in: *NeurIPS*, 2020, pp. 1–15.
- [326] S. Das, S. Tesfamariam, Y. Chen, Z. Qian, P. Tan, F. Zhou, Reliability-based optimization of nonlinear energy sink with negative stiffness and sliding friction, *Journal of Sound and Vibration* 485 (2020) 115560. doi:<https://doi.org/10.1016/j.jsv.2020.115560>.
- [327] M. Moustapha, Adaptive surrogate models for the reliable lightweight design of automotive body structures, Ph.D. thesis, Université Blaise Pascal-Clermont-Ferrand II, 2016.
- [328] M. Papadrakakis, N. D. Lagaros, Reliability-based structural optimization using neural networks and monte carlo simulation, *Computer methods in applied mechanics and engineering* 191 (2002) 3491–3507.
- [329] M. A. Bouhlel, J. T. Hwang, N. Bartoli, R. Lafage, J. Morlier, J. R. Martins, A python surrogate modeling framework with derivatives, *Advances in Engineering Software* 135 (2019) 102662.
- [330] M. Grieves, J. Vickers, Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems, in: *Transdisciplinary perspectives on complex systems*, Springer, 2017, pp. 85–113.
- [331] E. Glaessen, D. Stargel, The digital twin paradigm for future nasa and us air force vehicles, in: *53rd AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference 20th AIAA/ASME/AHS adaptive structures conference 14th AIAA*, 2012, p. 1818.
- [332] D. Jones, C. Snider, A. Nassehi, J. Yon, B. Hicks, Characterising the digital twin: A systematic literature review, *CIRP Journal of Manufacturing Science and Technology* 29 (2020) 36–52.
- [333] H. Van der Valk, H. Haße, F. Möller, M. Arbter, J.-L. Henning, B. Otto, A taxonomy of digital twins., in: *AMCIS*, 2020, pp. 1–10.
- [334] L. Wright, S. Davidson, How to tell the difference between a model and a digital twin, *Advanced Modeling and Simulation in Engineering Sciences* 7 (2020) 1–13.
- [335] D. Wagg, K. Worden, R. Barthorpe, P. Gardner, Digital twins: state-of-the-art and future directions for modeling and simulation in engineering dynamics applications, *ASCE-ASME J Risk and Uncert in Engrg Sys Part B Mech Engrg* 6 (2020).
- [336] H. Hinduja, S. Kekkar, S. Chourasia, H. B. Chakrapani, Industry 4.0: digital twin and its industrial applications, *Int. J. Sci. Eng. Technol. Open Access J* 8 (2020).
- [337] K. Y. H. Lim, P. Zheng, C.-H. Chen, A state-of-the-art survey of digital twin: techniques, engineering product lifecycle management and business innovation perspectives, *Journal of Intelligent Manufacturing* 31 (2020) 1313–1337.
- [338] F. Tao, F. Sui, A. Liu, Q. Qi, M. Zhang, B. Song, Z. Guo, S. C.-Y. Lu, A. Y. Nee, Digital twin-driven product design framework, *International Journal of Production Research* 57 (2019) 3935–3953. doi:[10.1080/00207543.2018.1443229](https://doi.org/10.1080/00207543.2018.1443229).
- [339] R. Stark, C. Fresemann, K. Lindow, Development and operation of digital twins for technical systems and services, *CIRP*



- Annals 68 (2019) 129–132.
- [340] A. Rasheed, O. San, T. Kvamsdal, Digital twin: Values, challenges and enablers, arXiv preprint arXiv:1910.01719 (2019).
- [341] J.-F. Uhlenkamp, K. Hribernik, S. Wellsandt, K.-D. Thoben, Digital twin applications: A first systemization of their dimensions, in: 2019 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC), IEEE, 2019, pp. 1–8. doi:10.1109/ICE.2019.8792579.
- [342] S. Haag, R. Anderl, Digital twin–proof of concept, Manufacturing Letters 15 (2018) 64–66.
- [343] T. Fei, C. Jiangfeng, Q. Qinglin, M. Zhang, H. Zhang, S. Fangyuan, Digital twin-driven product design, manufacturing and service with big data, The International Journal of Advanced Manufacturing Technology 94 (2018) 3563–3576.
- [344] B. R. Barricelli, E. Casiraghi, D. Fogli, A survey on digital twin: definitions, characteristics, applications, and design implications, IEEE access 7 (2019) 167653–167671.
- [345] R. Rosen, G. Von Wichert, G. Lo, K. D. Bettenhausen, About the importance of autonomy and digital twins for the future of manufacturing, IFAC-PapersOnLine 48 (2015) 567–572.
- [346] E. M. Kraft, The air force digital thread/digital twin-life cycle integration and use of computational and experimental knowledge, in: 54th AIAA aerospace sciences meeting, 2016, p. 0897.
- [347] M. Macchi, I. Roda, E. Negri, L. Fumagalli, Exploring the role of digital twin for asset lifecycle management, IFAC-PapersOnLine 51 (2018) 790–795.
- [348] J. Ghosh, S. Foulard, R. Fietzek, Vehicle mass estimation from CAN data and drivetrain torque observer, Technical Report, SAE Technical Paper, 2017.
- [349] P. Aivaliotis, K. Georgoulas, Z. Arkouli, S. Makris, Methodology for enabling digital twin using advanced physics-based modelling in predictive maintenance, Procedia Cirp 81 (2019) 417–422.
- [350] Y. Hua, X. Liu, S. Zhou, Y. Huang, H. Ling, S. Yang, Toward sustainable reuse of retired lithium-ion batteries from electric vehicles, Resources, Conservation and Recycling 168 (2021) 105249.
- [351] 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 Transactions on knowledge and data engineering 29 (2017) 2318–2331.
- [352] R. Wang, R. Yu, Physics-guided deep learning for dynamical systems: A survey, arXiv preprint arXiv:2107.01272 (2021).
- [353] M. E. Levine, A. M. Stuart, A framework for machine learning of model error in dynamical systems, arXiv preprint arXiv:2107.06658 (2021).
- [354] A. C. Miller, N. J. Foti, E. B. Fox, Breiman’s two cultures: You don’t have to choose sides, 2021. arXiv:2104.12219.
- [355] 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.
- [356] M. Stender, Data-driven techniques for the nonlinear dynamics of mechanical structures, Ph.D. thesis, Technische Universität Hamburg, 2020.
- [357] A. Karpatne, W. Watkins, J. Read, V. Kumar, Physics-guided neural networks (pgnn): An application in lake temperature modeling, arXiv preprint arXiv:1710.11431 (2017).
- [358] U. Forssell, P. Lindskog, Combining semi-physical and neural network modeling: An example of its usefulness, IFAC Proceedings Volumes 30 (1997) 767–770.
- [359] D. Kochkov, J. A. Smith, A. Alieva, Q. Wang, M. P. Brenner, S. Hoyer, Machine learning–accelerated computational fluid dynamics, Proceedings of the National Academy of Sciences 118 (2021).
- [360] Y. Yu, H. Yao, Y. Liu, Aircraft dynamics simulation using a novel physics-based learning method, Aerospace Science and Technology 87 (2019) 254–264.
- [361] J. N. Kani, A. H. Elsheikh, Dr-rnn: A deep residual recurrent neural network for model reduction, arXiv preprint arXiv:1709.00939 (2017).
- [362] A. Gupta, P. F. J. Lermusiaux, Neural closure models for dynamical systems, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 477 (2021) 20201004. doi:10.1098/rspa.2020.1004.
- [363] J. Wilson, L. Zorzetto, A generalised approach to process state estimation using hybrid artificial neural network/mechanistic models, Computers & chemical engineering 21 (1997) 951–963.
- [364] E. J. Parish, K. Duraisamy, A paradigm for data-driven predictive modeling using field inversion and machine learning, Journal of Computational Physics 305 (2016) 758–774.
- [365] A. P. Singh, S. Medida, K. Duraisamy, Machine-learning-augmented predictive modeling of turbulent separated flows over airfoils, AIAA journal 55 (2017) 2215–2227.
- [366] J. Buist, B. Sanderse, Y. van Halder, B. Koren, G. van Heijst, Machine learning for closure models in multiphase-flow applications, Crete, Greece (2019).
- [367] B. D. Tracey, K. Duraisamy, J. J. Alonso, A machine learning strategy to assist turbulence model development, in: 53rd AIAA aerospace sciences meeting, 2015, p. 1287.
- [368] X. Jia, J. Willard, A. Karpatne, J. Read, J. Zwart, M. Steinbach, V. Kumar, Physics guided rnns for modeling dynamical systems: A case study in simulating lake temperature profiles, in: Proceedings of the 2019 SIAM International Conference on Data Mining, SIAM, 2019, pp. 558–566.
- [369] X. Jia, J. Willard, A. Karpatne, J. S. Read, J. A. Zwart, M. Steinbach, V. Kumar, Physics-guided machine learning for scientific discovery: An application in simulating lake temperature profiles, ACM/IMS Transactions on Data Science 2 (2021) 1–26.
- [370] J. S. Read, X. Jia, J. Willard, A. P. Appling, J. A. Zwart, S. K. Oliver, A. Karpatne, G. J. Hansen, P. C. Hanson, W. Watkins, et al., Process-guided deep learning predictions of lake water temperature, Water Resources Research 55 (2019) 9173–9190.
- [371] A. Daw, R. Q. Thomas, C. C. Carey, J. S. Read, A. P. Appling, A. Karpatne, Physics-guided architecture (pga) of neural networks for quantifying uncertainty in lake temperature modeling, in: Proceedings of the 2020 siam international conference on data mining, SIAM, 2020, pp. 532–540.
- [372] R. Zhang, Y. Liu, H. Sun, Physics-guided convolutional neural network (phycnn) for data-driven seismic response modeling, Engineering Structures 215 (2020) 110704. doi:https://doi.org/10.1016/j.engstruct.2020.110704.
- [373] S. Karimpouli, P. Tahmasebi, Physics informed machine learning: Seismic wave equation, Geoscience Frontiers 11 (2020) 1993–2001. doi:https://doi.org/10.1016/j.gsf.2020.07.007.
- [374] J. Sun, K. A. Innanen, C. Huang, Physics-guided deep learning for seismic inversion with hybrid training and uncertainty analysis, Geophysics 86 (2021) R303–R317.
- [375] M. Raissi, P. Perdikaris, G. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, Journal of Computational Physics 378 (2019) 686–707. doi:https://doi.org/10.1016/j.jcp.2018.10.045.
- [376] A. D. Jagtap, E. Kharazmi, G. E. Karniadakis, Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward

- and inverse problems, *Computer Methods in Applied Mechanics and Engineering* 365 (2020) 113028. doi:<https://doi.org/10.1016/j.cma.2020.113028>.
- [377] G. Pang, L. Lu, G. E. Karniadakis, fpinns: Fractional physics-informed neural networks, *SIAM Journal on Scientific Computing* 41 (2019) A2603–A2626.
- [378] Z. Mao, A. D. Jagtap, G. E. Karniadakis, Physics-informed neural networks for high-speed flows, *Computer Methods in Applied Mechanics and Engineering* 360 (2020) 112789. doi:<https://doi.org/10.1016/j.cma.2019.112789>.
- [379] X. Meng, Z. Li, D. Zhang, G. E. Karniadakis, Ppinn: Parareal physics-informed neural network for time-dependent pdes, *Computer Methods in Applied Mechanics and Engineering* 370 (2020) 113250. doi:<https://doi.org/10.1016/j.cma.2020.113250>.
- [380] X. Jin, S. Cai, H. Li, G. E. Karniadakis, Nsfnets (navier-stokes flow nets): Physics-informed neural networks for the incompressible navier-stokes equations, *Journal of Computational Physics* 426 (2021) 109951.
- [381] D. Liu, Y. Wang, Multi-fidelity physics-constrained neural network and its application in materials modeling, *Journal of Mechanical Design* 141 (2019).
- [382] Y. Chen, L. Lu, G. E. Karniadakis, L. Dal Negro, Physics-informed neural networks for inverse problems in nano-optics and metamaterials, *Optics express* 28 (2020) 11618–11633.
- [383] Z. Fang, J. Zhan, Deep physical informed neural networks for metamaterial design, *IEEE Access* 8 (2019) 24506–24513.
- [384] M. Raissi, H. Babaei, P. Givi, Deep learning of turbulent scalar mixing, *Physical Review Fluids* 4 (2019) 124501.
- [385] H. Guo, X. Zhuang, T. Rabczuk, A deep collocation method for the bending analysis of kirchhoff plate, *arXiv preprint arXiv:2102.02617* (2021).
- [386] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, J. M. Siskind, Automatic differentiation in machine learning: a survey, *Journal of machine learning research* 18 (2018).
- [387] X. Meng, G. E. Karniadakis, A composite neural network that learns from multi-fidelity data: Application to function approximation and inverse pde problems, *Journal of Computational Physics* 401 (2020) 109020. doi:<https://doi.org/10.1016/j.jcp.2019.109020>.
- [388] L. Lu, P. Jin, G. E. Karniadakis, Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators, *arXiv preprint arXiv:1910.03193* (2019).
- [389] J. Berg, K. Nyström, Data-driven discovery of pdes in complex datasets, *Journal of Computational Physics* 384 (2019) 239–252.
- [390] A. M. Tartakovsky, C. O. Marrero, P. Perdikaris, G. D. Tartakovsky, D. Barajas-Solano, Learning parameters and constitutive relationships with physics informed deep neural networks, *arXiv preprint arXiv:1808.03398* (2018).
- [391] Y. Zhu, N. Zabarav, P.-S. Koutsourelakis, 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. doi:<https://doi.org/10.1016/j.jcp.2019.05.024>.
- [392] Y. Yang, P. Perdikaris, Physics-informed deep generative models, 2018. *arXiv:1812.03511*.
- [393] B. Lütjens, C. H. Crawford, M. Veillette, D. Newman, Spectral pinns: Fast uncertainty propagation with physics-informed neural networks, in: *35th Conference on Neural Information Processing Systems*, 2021, pp. 1–10.
- [394] Y. Gao, M. K. Ng, Wasserstein generative adversarial uncertainty quantification in physics-informed neural networks, 2021. *arXiv:2108.13054*.
- [395] D. Zhang, L. Lu, L. Guo, G. E. Karniadakis, Quantifying total uncertainty in physics-informed neural networks for solving forward and inverse stochastic problems, *Journal of Computational Physics* 397 (2019) 108850. doi:<https://doi.org/10.1016/j.jcp.2019.07.048>.
- [396] L. Yang, X. Meng, G. E. Karniadakis, B-pinns: Bayesian physics-informed neural networks for forward and inverse pde problems with noisy data, *Journal of Computational Physics* 425 (2021) 109913. doi:<https://doi.org/10.1016/j.jcp.2020.109913>.
- [397] J. Chen, Y. Liu, Probabilistic physics-guided machine learning for fatigue data analysis, *Expert Systems with Applications* 168 (2021) 114316. doi:<https://doi.org/10.1016/j.eswa.2020.114316>.
- [398] N. Muralidhar, J. Bu, Z. Cao, L. He, N. Ramakrishnan, D. Tafti, A. Karpatne, Physics-guided design and learning of neural networks for predicting drag force on particle suspensions in moving fluids, *arXiv preprint arXiv:1911.04240* (2019).
- [399] S. Saemundsson, A. Terenin, K. Hofmann, M. P. Deisenroth, Variational integrator networks for physically structured embeddings, 2020. *arXiv:1910.09349*.
- [400] I. Huh, E. Yang, S. J. Hwang, J. Shin, Time-reversal symmetric ode network, 2021. *arXiv:2007.11362*.
- [401] A. Botev, A. Jaegle, P. Wirsberger, D. Hennes, I. Higgins, Which priors matter? benchmarking models for learning latent dynamics, *arXiv preprint arXiv:2111.05458* (2021).
- [402] E. Dupont, A. Doucet, Y. W. Teh, Augmented neural odes, *arXiv preprint arXiv:1904.01681* (2019).
- [403] S. Massaroli, M. Poli, J. Park, A. Yamashita, H. Asama, Dissecting neural odes, *arXiv preprint arXiv:2002.08071* (2020).
- [404] M. Lutter, J. Peters, Combining physics and deep learning to learn continuous-time dynamics models, *arXiv preprint arXiv:2110.01894* (2021).
- [405] S. Greydanus, M. Dzamba, J. Yosinski, Hamiltonian neural networks, 2019. *arXiv:1906.01563*.
- [406] Y. D. Zhong, B. Dey, A. Chakraborty, Symplectic ode-net: Learning hamiltonian dynamics with control, *arXiv preprint arXiv:1909.12077* (2019).
- [407] Y. D. Zhong, B. Dey, A. Chakraborty, Dissipative symoden: Encoding hamiltonian dynamics with dissipation and control into deep learning, *arXiv preprint arXiv:2002.08860* (2020).
- [408] Z. Chen, J. Zhang, M. Arjovsky, L. Bottou, Symplectic recurrent neural networks, *arXiv preprint arXiv:1909.13334* (2019).
- [409] M. Cranmer, S. Greydanus, S. Hoyer, P. Battaglia, D. Spergel, S. Ho, Lagrangian neural networks, 2020. *arXiv:2003.04630*.
- [410] M. Lutter, C. Ritter, J. Peters, Deep lagrangian networks: Using physics as model prior for deep learning, *arXiv preprint arXiv:1907.04490* (2019).
- [411] M. Finzi, K. A. Wang, A. G. Wilson, Simplifying hamiltonian and lagrangian neural networks via explicit constraints, *arXiv preprint arXiv:2010.13581* (2020).
- [412] R. Bhattoo, S. Ranu, N. Krishnan, Lagrangian neural network with differentiable symmetries and relational inductive bias, *arXiv preprint arXiv:2110.03266* (2021).
- [413] Y. D. Zhong, B. Dey, A. Chakraborty, Benchmarking energy-conserving neural networks for learning dynamics from data, in: A. Jadbabaie, J. Lygeros, G. J. Pappas, P. A. Parrilo, B. Recht, C. J. Tomlin, M. N. Zeilinger (Eds.), *Proceedings of the 3rd Conference on Learning for Dynamics and Control*, volume 144 of *Proceedings of Machine Learning Research*, PMLR, 2021, pp. 1218–1229.
- [414] Y. Ba, G. Zhao, A. Kadambi, Blending diverse physical priors with neural networks, *arXiv preprint arXiv:1910.00201*

- (2019).
- [415] S. Pawar, O. San, B. Aksoylu, A. Rasheed, T. Kvamsdal, Physics guided machine learning using simplified theories, *Physics of Fluids* 33 (2021) 011701.
- [416] K. Kashinath, M. Mustafa, A. Albert, J. Wu, C. Jiang, S. Esmailzadeh, K. Azizzadenesheli, R. Wang, A. Chattopadhyay, A. Singh, et al., Physics-informed machine learning: case studies for weather and climate modelling, *Philosophical Transactions of the Royal Society A* 379 (2021) 20200093.
- [417] E. Kodra, U. Bhatia, S. Chatterjee, S. Chen, A. R. Ganguly, Physics-guided probabilistic modeling of extreme precipitation under climate change, *Scientific reports* 10 (2020) 1–11.
- [418] T. Beucler, S. Rasp, M. Pritchard, P. Gentine, Achieving conservation of energy in neural network emulators for climate modeling, *arXiv preprint arXiv:1906.06622* (2019).
- [419] S. Kumar, S. Tan, L. Zheng, D. M. Kochmann, Inverse-designed spinodoid metamaterials, *npj Computational Materials* 6 (2020) 1–10.
- [420] R. Cang, H. Li, H. Yao, Y. Jiao, Y. Ren, Improving direct physical properties prediction of heterogeneous materials from imaging data via convolutional neural network and a morphology-aware generative model, *Computational Materials Science* 150 (2018) 212–221.
- [421] Y. Yu, H. Yao, Y. Liu, Structural dynamics simulation using a novel physics-guided machine learning method, *Engineering Applications of Artificial Intelligence* 96 (2020) 103947. doi:<https://doi.org/10.1016/j.engappai.2020.103947>.
- [422] Y. D. Zhong, B. Dey, A. Chakraborty, Extending lagrangian and hamiltonian neural networks with differentiable contact models, *Advances in Neural Information Processing Systems* 34 (2021).
- [423] Y. Yin, V. L. Guen, J. Dona, E. de Bézenac, I. Ayed, N. Thome, P. Gallinari, Augmenting physical models with deep networks for complex dynamics forecasting, *arXiv preprint arXiv:2010.04456* (2020).
- [424] P. Friederich, M. Krenn, I. Tamblyn, A. Aspuru-Guzik, Scientific intuition inspired by machine learning-generated hypotheses, *Machine Learning: Science and Technology* 2 (2021) 025027.
- [425] S. J. Wetzel, R. G. Melko, J. Scott, M. Panju, V. Ganesh, Discovering symmetry invariants and conserved quantities by interpreting siamese neural networks, *Physical Review Research* 2 (2020) 033499.
- [426] R. Iten, T. Metger, H. Wilming, L. Del Rio, R. Renner, Discovering physical concepts with neural networks, *Physical review letters* 124 (2020) 010508.
- [427] Z. Lai, C. Mylonas, S. Nagarajaiah, E. Chatzi, Structural identification with physics-informed neural ordinary differential equations, *Journal of Sound and Vibration* 508 (2021) 116196. doi:<https://doi.org/10.1016/j.jsv.2021.116196>.
- [428] P. Toth, D. J. Rezende, A. Jaegle, S. Racanière, A. Botev, I. Higgins, Hamiltonian generative networks, 2020. *arXiv:1909.13789*.
- [429] M. A. Roehrl, T. A. Runkler, V. Brandstetter, M. Tokic, S. Obermayer, Modeling system dynamics with physics-informed neural networks based on lagrangian mechanics, *IFAC-PapersOnLine* 53 (2020) 9195–9200.
- [430] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, S. Wermter, Continual lifelong learning with neural networks: A review, *Neural Networks* 113 (2019) 54–71. doi:<https://doi.org/10.1016/j.neunet.2019.01.012>.
- [431] M. A. Osborne, R. Garnett, S. J. Roberts, Gaussian processes for global optimization, in: *3rd international conference on learning and intelligent optimization (LION3)*, Citeseer, 2009, pp. 1–15.
- [432] J. Fu, K. Luo, S. Levine, Learning robust rewards with adversarial inverse reinforcement learning, *arXiv preprint arXiv:1710.11248* (2017).
- [433] X. Chen, E. Minasyan, J. D. Lee, E. Hazan, Provable regret bounds for deep online learning and control, *arXiv preprint arXiv:2110.07807* (2021).
- [434] D. Wang, H. He, D. Liu, Adaptive critic nonlinear robust control: A survey, *IEEE transactions on cybernetics* 47 (2017) 3429–3451.