

Chapter 1

Machine Learning in Computer Aided Engineering



Francisco J. Montáns, Elías Cueto, and Klaus-Jürgen Bathe

1.1 Introduction

The purpose of Machine Learning algorithms is to learn automatically from data employing general procedures. Machine Learning (ML) is today ubiquitous due to its success in many current daily applications such as face recognition (Hassan and Abdulazeez 2021), speech (Malik et al. 2021) and speaker recognition (Hanifa et al. 2021), credit card fraud detection (Ashtiani and Raahemi 2021; Nayak et al. 2021), spam detection (Akinyelu 2021), and cloud security (Nassif et al. 2021). ML governs our specific Google searches and the advertisements we receive (Kim et al. 2001) based on our past actions, along many other interactions (Google cloud 2023). It even anticipates what we will type or what we will do. And, of course, ML schemes also rank us, scientists (Beel and Gipp 2009).

The explosion of applications of ML came with the increased computer power and also the ubiquitous presence of computers, cell phones, and other “smart” devices. These gave ML the spotlight to foster its widespread use to many other areas in which

F. J. Montáns (✉)

E.T.S. de Ingeniería Aeronáutica y del Espacio, Universidad Politécnica de Madrid, Pza. Cardenal Cisneros 3, 28040 Madrid, Spain

e-mail: fco.montans@upm.es

Department of Mechanical and Aerospace Engineering, Herbert Wertheim College of Engineering, University of Florida, Gainesville, FL 32611, USA

E. Cueto

Aragon Institute of Engineering Research, Universidad de Zaragoza, Maria de Luna s/n, 50018 Zaragoza, Spain

e-mail: ecueto@unizar.es

K. J. Bathe

Mechanical Engineering Department, Massachusetts Institute of Technology, 77 Mass. Ave., Cambridge, MA 02139, USA

e-mail: kjb@mit.edu

© The Author(s) 2023

T. Rabczuk and K. J. Bathe (eds.), *Machine Learning in Modeling and Simulation*, Computational Methods in Engineering & the Sciences,

https://doi.org/10.1007/978-3-031-36644-4_1

it had less presence. The success in many extremely useful areas such as speech and face recognition has contributed to this interest (Marr 2019). Today, ML may help you (through web services) to find a job, obtain a loan, find a partner, obtain an insurance, and, among others, also helps in the medical and legal services (Duarte 2018). Of course, ML raises many ethical issues, some of which are described, for example in Stahl (2021). However, the discovered power and success of ML in many areas have made a very important impact on our society and, remarkably, on how many problems are addressed. No wonder, the number of ML papers published in almost all fields has sharply increased in the last 10 years, with a rate following approximately Moore's law (Frank et al. 2020).

Machine Learning is considered a part of Artificial Intelligence (AI) (Michalski et al. 2013). In essence, ML algorithms are general procedures and codes that, with the information from datasets, can give predictions for a wide range of problems (see Fig. 1.1). The main difference to classical programs is that the classical programs are developed for specific applications, like in Computer Aided Engineering, which is the topic of this chapter, to solve specific differential equations in integral forms. An example is how finite elements have been developed. In contrast, ML procedures are for much more general applications, being used almost unchanged in problems apparently unconnected as predicting the evolutions of stocks, spam filtering, face recognition, typing prediction, pharmacologic design, or materials selection. ML methods are different also from Expert Systems because these are based on fixed rules or fixed probability structures. ML methods excel when useful information needs to be obtained from massive amounts of data.

Of course, generality comes usually with a trade-off regarding efficiency for a specific problem solution (Fig. 1.1), so the use of ML for the solution of simple problems, or for problems which can be solved by other more specific procedures, is typically inappropriate. Furthermore, ML is used when predictions are needed for problems which have not or cannot be accurately formulated; that is, when the variables and mathematical equations governing the problem are not fully determined—but physics-informed approaches with ML are now also much focused on, Raissi et al. (2019). Nonetheless, ML codes and procedures are still mostly used as general “black boxes”, typically employing standard implementations available in free and open-source software repositories. A number of input variables are then employed and some specific output is desired, which together comprises the input-to-output process learned and adjusted from known cases or from the structure of the input data. Some of these free codes are Scikit-learn (Pedregosa et al. 2011) (one of the best-known), Microsoft Cognitive Toolkit (Xiong et al. 2018), TensorFlow (Dillon et al. 2017) (which is optimal for Cuda-enabled Graphic Processing Unit (GPU) parallel ML), Keras (Gulli and Pal 2017), OpenNN (Build powerful models 2022), and SystemML (Ghoting et al. 2011), just to name a few. Other proprietary software, used by big companies, are AWS Machine Learning Services from Amazon (Hashemipour and Ali 2020), Cloud Machine Learning Engine from Google (Bisong 2019a), Matlab (Paluszek and Thomas 2016; Kim 2017), Mathematica (Brodie et al. 2020; Rodríguez and Kramer 2019), etc. Moreover, many software offerings have libraries for ML, and are often used in ML projects like Python (NumPy, Bisong

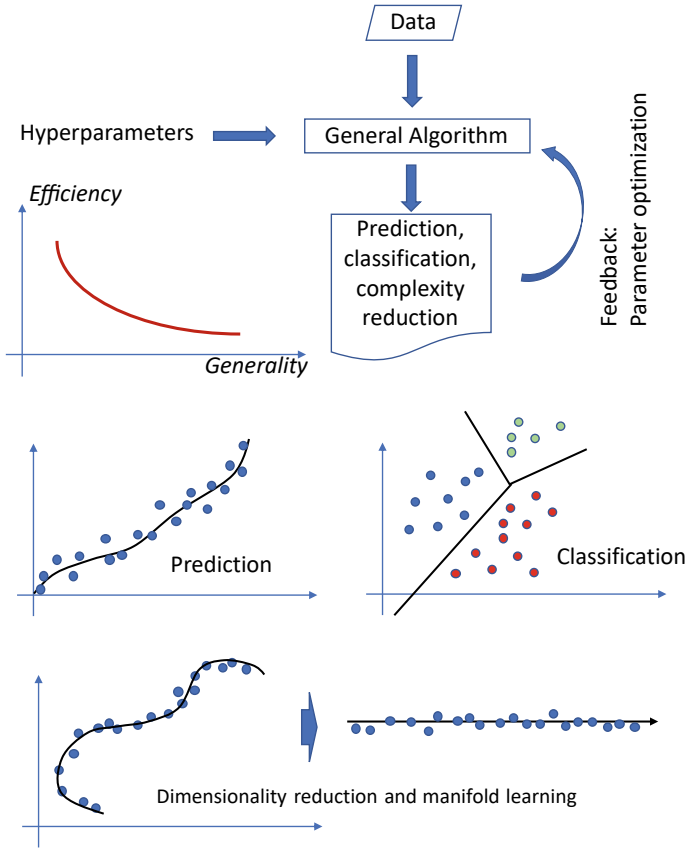


Fig. 1.1 Overall Machine Learning (ML) process and contrast between efficiency and generality of the method. Hyperparameters are user-defined parameters which account for the type of problem, whereas parameters are optimized for best prediction. ML may be used for prediction and for classification. it is also often used as a tool for dimensionality reduction

2019b, Scikit-learn, Pedregosa et al. 2011, and Tensorly, Kossaifi et al. 2016; see review in Stančin and Jović 2019), C++, e.g., Kaehler and Bradski (2016), Julia (a recent Just-In-Time (JIT) compiling language created with science and ML in mind, Gao et al. 2020; Innes 2018; Innes et al. 2019), and the R programming environment (Lantz 2019; Bischl et al. 2016; Molnar et al. 2018); see also Raschka and Mirjalili (2019), King (2009), Gao et al. (2020), Bischl et al. (2016). These software offerings also use many earlier published methods for standard computational tasks such as mathematical libraries (like for curve fitting, the solution of linear and nonlinear equations, the determination of eigenvalues and eigenvectors or Singular Value Decompositions), and computational procedures for optimization (e.g., the steepest descent algorithms). The offerings also use earlier established statistical and regression algorithms, interpolation, clustering, domain slicing (e.g., tessellation algorithms), and function approximations.

ML derives from the conceptually fuzzy (uncertain, non-deterministic) learning approach of AI. AI is devoted to mimicking the way the human learning process works—namely, the human brain, through the establishment of neurological connections based on observations, can perform predictions, albeit mostly only qualitative, of new events. And then, the more experience (data) has been gathered, the better are the predictions through experience reinforcements and variability of observations. In addition, classification is another task typically performed by the human brain. We classify photos, people, experiences, and so on, according to some common features: we search continuously for features that allow us to group and separate out things so that we can establish relations of outcomes to such groups. Abundant data, data structuring, and data selection and simplification are crucial pieces of this type of “fuzzy” learning and, hence, of ML procedures.

Based on these observations, neural network concepts were rather early developed by McCulloch and Pitts in 1943 and Hebb in 1949 (Hebb 2005), who wrote the well-known sentence “Cells that fire together, wire together”, meaning that the firing of one cell determines the actions of subsequent cells. While Hebb’s forward firing rule is unstable through successive epochs, it was the foundation for Artificial Neural Network (NN) theories. Probably due to the difficulties in implementations and computational cost in using NN, the widespread use of NN was delayed until the 1990s. The introduction of improvements in the procedures for backpropagation and optimization, as well as improvements in data acquisition, information retrieval, and data mining, made possible the application of NNs to real problems. Today, NNs are very flexible and are the basis of many ML techniques and applications. However, this delay also facilitated the appearance and use of other ML-related methods as expert systems and decision trees, and a myriad of pattern recognition and decision-making approaches.

Today, whenever a complex problem is found, especially if there is no sound theory or reliable formulation to solve it, ML is a valuable tool to try. In many cases, the result is successful and indeed even a good understanding of the behavior of the problem and the variables involved may be obtained. While the introduction of ML procedures into Computer Aided Engineering (CAE) took a longer time than in other areas, probably because for many problems the governing equations and effective computational procedures were known, ML is finally also focused on addressing complex and computationally intensive CAE solutions. In this chapter, we overview some of the procedures and applications of Machine Learning employed in CAE.

1.2 Machine Learning Procedures Employed in CAE

As mentioned, ML is often considered to be a subset of AI (Michalski et al. 2013; Dhanalaxmi 2020; Karthikeyan et al. 2021), although often ML is also recognized as a separate field itself which only has some intersection with AI (Manavalan 2020; Langley 2011; Ongsulee 2017). Deep Learning (DL) is a subset of ML. Although the use of NNs is the most common approach to address CAE problems and ML

problems in general, there are many other ML techniques that are being used. We review below the fundamental aspects of these techniques.

1.2.1 Machine Learning Aspects and Classification of Procedures

Our objective in this section is to focus on various fundamental procedures commonly used in ML schemes.

1.2.1.1 Classification, Identification, and Prediction

ML procedures are mainly employed for three tasks: classification, identification (both may broadly be considered as classification), and prediction. An example of classification is the labeling of e-mails as spam or not spam (Gaurav et al. 2020; Crawford et al. 2015). Examples of identification are the identification of a type of behavior or material from some stress–strain history or from force signals in machining (Denkena et al. 2019; Penumuru et al. 2020; Bock et al. 2019), the identification of a nanostructure from optical microscopy (Lin et al. 2018), the identification of a person from a set of images (Ahmed et al. 2015; Ding et al. 2015; Sharma et al. 2020), and the identification of a sentence from some fuzzy input. Examples of prediction are the prediction of behavior of a material under some deformation pattern (Ye et al. 2022; Ibragimova et al. 2021; Huang et al. 2020), the prediction of a sentence from some initial words (Bickel et al. 2005; Sordoni et al. 2015), and the prediction of the trajectory of salient flying objects (Wu et al. 2017; Fu et al. 2020). Of course, there are some AI procedures which may not only belong to one of these categories, as the identification or prediction of governing equations in physics (Rai and Sahu 2020; Raissi and Karniadakis 2018). Clustering ML procedures are typically used for classification, whereas regression ML procedures are customarily used for prediction.

1.2.1.2 Expected and Unexpected Data Relations

Another relevant distinction is between ML approaches and Data Mining (DM). ML focuses on using known properties of data in classification or in prediction, whereas DM focuses on the discovery of new unknown properties or relations of data. However, ML, along information systems, is often considered part of DM (Adriaans and Zantinge 1997). The overlap of DM and ML is seen in cases like the discovery of unknown relations or in finding optimum state variables which may be, for example, given in physical equations. Note that ML typically assumes that we know beforehand the existence of relations (e.g., which are the relevant variables and what is the type of output we expect), but the purpose of DM is to research the existence of perhaps unexpected relations from raw data.

1.2.1.3 Statistical and Optimization Approaches within ML

Many procedures use, or are derived from, statistics, and in particular probability theory (Murphy 2012; Bzdok et al. 2018). In a similar manner, ML employs mostly optimization procedures (Le et al. 2011). The main conceptual difference between these theories and ML is the purpose of the developments. In the case of statistics, the purpose is to obtain inference or characteristics of the population such as the distribution and the mean (which of course could be used thereafter for predictions); see Fig. 1.2. In the case of ML, the purpose is to predict new outcomes, often without the need for statistically characterizing populations, and incorporate these outcomes in further predictions (Bzdok et al. 2018). ML approaches often support predictions on models. ML optimizes parameters for obtaining the best predictions as quantified by a cost function, and the values of these parameters are optimized also to account for the uncertainty in the data and in the predictions. ML approaches may use statistical distributions, but those are not an objective and their evaluation is often numerical (ML is interested in predictions). Also, while ML uses optimization procedures to obtain values of parameters, the objective is not to obtain the “optimum” solution to fit data, but a parsimonious model giving reliable predictions (e.g., to avoid overfitting).

1.2.1.4 Supervised, Unsupervised, and Reinforced Learning

It is typical to classify the ML procedures in supervised, unsupervised, semi-supervised, and reinforced learning (Raschka 2015; Burkov 2019, 2020).

In **supervised learning**, samples $\{s_i \equiv \{\mathbf{x}_i, y_i\}\}_{i=1,\dots,n} \in \mathcal{S}$ with vectors of features \mathbf{x}_i are labeled with a known result or label y_i . The label may be a class, a

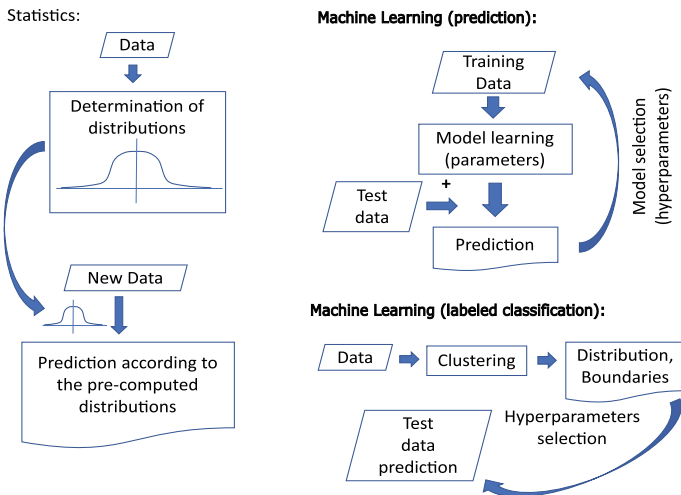


Fig. 1.2 Comparison of classical statistics with machine learning approaches

number, a matrix, or other. The purpose of the ML approach in this case is (typically) to create a model that relates those known outputs y_i to the dataset samples through some combination of the $j = 1, \dots, N$ features $x_{j(i)} \equiv x_{ji}$ of each sample i . $x_{j(i)}$ are also referred to as data, variables, measurements, or characteristics, depending on the context or field of application. An example of a ML procedure could be to relate the seismic vulnerability of a building (label) as a function of features like construction type, age, size, location, building materials, maintenance, etc. Rosti et al. (2022), Zhang et al. (2019), Ruggieri et al. (2021). The ML purpose is here to be able to learn the vulnerability of buildings from known vulnerabilities of other buildings. The labeling could have been obtained from experts or from past earthquakes. Supervised learning is based on sufficient known data, and we want to determine predictions in the nearby domain. In essence, we can say that “supervised learning is a high-dimensional interpolation problem” (Mallat 2016; Gin et al. 2021). We note that supervised learning may be improved with further data when available, since it is a dynamic learning procedure, mimicking the human brain.

In **unsupervised learning** the samples s_i are unlabeled ($s_i \equiv \{\mathbf{x}_i\}$), so the purpose is to label the samples from learning similitudes and common characteristics in the features of the samples; it is usually an instance-based learning. Typical unsupervised ML approaches are employed in clustering (e.g., classifying the structures by type in our previous example), dimensionality reduction (detecting which features are less relevant to the output label, for example because all or most samples have it, like doors in buildings), and outlier detection (e.g., in detecting abnormal traffic in the Internet, Salman et al. 2020, 2022; Salloum et al. 2020) for the case when very few samples have that feature. These approaches are similar to data mining.

Semi-supervised learning is conceptually a combination of the previous approaches but with specific ML procedures. In essence it is a supervised learning approach in which there are few labeled samples (output known) but many more unlabeled samples (output unknown), even sometimes with incomplete features, with some missing characteristics, which may be filled in by imputation techniques (Lakshminarayan et al. 1996; Ramoni and Sebastiani 2001; Liu et al. 2012; Rabin and Fishelov 2017). The point here is that by having many more samples with unassigned features, we can determine better the statistical distributions of the data and the possible significance of the features in the result, resulting in an improvement over using only labeled data for which the features have been used to determine the label. For example, in our seismic vulnerability example, imagine that one feature is that the building has windows. Since almost all buildings have windows, it is unlikely that this feature is relevant in determining the vulnerability (it will give little Information Gain; see below). On the contrary, if 20% of the buildings have a steel structure, and if the correlation is positive regarding the (lack of) vulnerability, it is likely that the feature is important in determining the vulnerability.

There is also another type of ML seldom used in CAE, which is **reinforced learning** (or reward-based learning). In this case, the computer develops and changes actions to learn a policy depending on the feedback, i.e. rewards which themselves modify the subsequent actions by maximizing the expected reward. It has some common concepts to supervised learning, but the purpose is an action, instead of a

prediction. Hence, it is a typical ML approach in control dynamics (Buşoniu et al. 2018; Lewis and Liu 2013) with applications, for example, in the aeronautical industry (Choi and Cha 2019; Swischuk and Allaire 2019; He et al. 2021).

1.2.1.5 Data Cleaning, Ingestion, Augmentation, Curation, Data Evaluation, and Data Standardization

Data is the key to ML procedures, so datasets are usually large and obtained in different ways. The importance of data requires that the data is presented to the ML method (and maintained if applicable) in optimal format. To reach that goal requires many processes which often also involve ML techniques. For example, in a dataset there may be data which are not in a logical range, or with missing entries, hence they need to be **cleaned**. ML techniques may be used to determine outliers in datasets, or assign values (*data imputation*) according to the other features and labels present in other samples in the dataset. Different dataset formats such as qualitative entries like “good”, “fair”, or “bad”, and quantitative entries like “1–9”, may need to be converted (*encoded*) to standardized formats, using also ML algorithms (e.g., assigning “fair” to a numerical value according to samples in the dataset). This is called **data ingestion**. ML procedures may also need to have data distributions determined, that is, **data evaluated** to learn if a feature follows a normal distribution or if there is a consistent bias, and also **standardize data** according to min–max values or the same normal distribution, for example to avoid numerical issues and give proper weight to different features. In large dynamic databases, much effort is expended for the proper maintenance of the data so it remains useful, using many operations such as data cleaning, organization, and labeling. This is called **data curation**.

Another aspect of data treatment is the creation of a **training set**, a **validation set**, and a **test set** from a database (although often test data refers to both the validation and the test set, in particular when only one model is considered). The purpose of the training set is to train the ML algorithm: to create the “model”. The purpose of the validation set is to evaluate the models in an independent way from the training set, for example to see which hyperparameters are best suited, or even which ML method is best suited. Examples may be the number of neurons in a neural network or the smoothing hyperparameter in splines fitting; different smoothing parameters yield different models for the same training set, and the validation set helps to select the best values, obtaining the best predictions but avoiding overfitting. Recall that ML is not interested in the minimum error for the training set, but in a predictive reliable model. The test set is used to evaluate the performance of the *final* selected model from the overall learning process. An accurate prediction of the training set with a poor prediction of the test set is an indicator of overfitting: we have reached an unreliable model. A model with similar accuracy in the training and test sets is a good model. The training set should not be used for assessing the accuracy of the model because the parameters and their values have been selected based on these data and hence overfitting may not be detected. However, if more data is needed for training, there are techniques for **data augmentation**, typically performing variations, transformations,

or combinations of other data (Shorten and Khoshgoftaar 2019). A typical example is to perform transformations of images (rotations, translations, changes in light, etc., Inoue 2018). Data augmentation should be used with care, because there is a risk that the algorithms correlate unexpected features with outputs: samples obtained by augmentation may have repetitive features because in the end they are correlated samples. These repetitive features may mislead the algorithms so they identify the feature as a key aspect to correlate to the output (Rice et al. 2020). An example is a random spot in an image that is being used for data augmentation. If the spot is present in the many generated samples, it may be correlated to the output as an important feature.

1.2.1.6 Overfitting, Regularization, and Cross-Validation

Overfitting and model complexity are important aspects in ML; see Fig. 1.3. Given that the data has errors and often some stochastic nature, a model which gives zero error in the training data does not mean that it is a good model; indeed it is usually a hint that it is the opposite: a presentation of overfitting (Fig. 1.3a). Best models are those less complex (parsimonious) models that follow Occam’s razor. They are as simple as possible but still have a great predictive power. Hence, the less parameters, the better. However, it is often difficult to simplify ML models to have few “smart” parameters, so model reduction and regularization techniques are often used as a “no-brainer” remedy for overfitting. Typical regularization (“smoothing”) techniques are Least Absolute Shrinkage Selection Operator, sparse or L1 regularization (LASSO) (Xu et al. 2008) and L2, called Ridge (Tikhonov 1963), or noise (Bishop 1995) regularization, or regression. The LASSO scheme “shrinks” the less important features

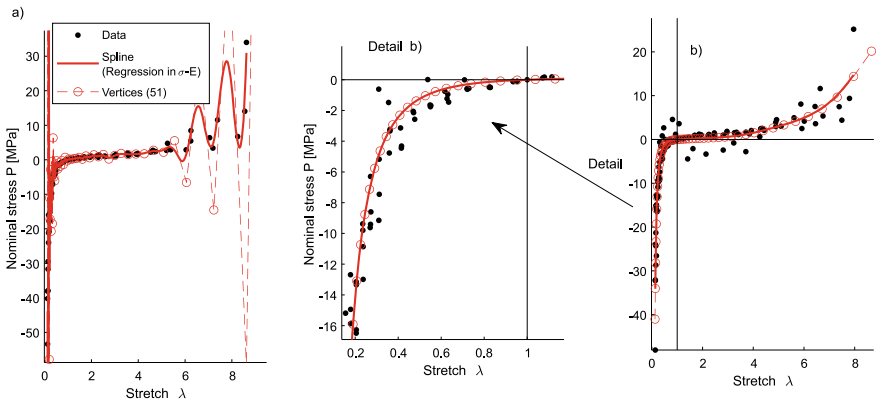


Fig. 1.3 Using B-splines to fit hyperelastic stress–strain data. Regression may be performed in nominal stress–stretch ($P - \lambda$) axes, or in true stress–strain $\sigma - E$ axes; note that the result is different. While usual test representation in hyperelasticity is in the ($P - \lambda$) axes, regression is preferred in $\sigma - E$ because of the symmetry of tension and compression in logarithmic strains). B-spline fit of experimental data with **a** overfitting and **b** proper fit using regularization based on stability conditions. Modified from Latorre and Montáns (2020)

(hence is also used for feature selection), whereas the L2 scheme gives a more even weight to them. The combination of both is known as *elastic net regularization* (Zou and Hastie 2005).

Model selection taking into account model fitness and including a penalization for model complexity is often performed by employing the Akaike Information Criterion (AIC). Given a collection of models arising from the available data, the AIC allows to compare these models among them, so as to help select the best fitted model. In essence, the AIC not only estimates the relative amount of information lost by each model but also takes into account its parsimony. In other words, it deals with the trade-off between overfitting and underfitting by computing

$$AIC = 2p - 2 \ln \mathcal{L} \tag{1.1}$$

where p is the number of parameters of the model (complexity penalty) and \mathcal{L} is the maximum of the likelihood function of the model, the joint probability of the observed data as a function of the p parameters of the model (see next section). Therefore, the chosen model should be the one with the minimum AIC. In essence, the AIC penalizes the number of parameters to choose the best model—and that is the model not only with as few parameters as possible but also with a large probability of reproducing the data using these parameters.

Dividing the data into two sets, one for training and one for validation, very often produces overfitting, especially for small datasets. To avoid the overfitting, the method of k -fold cross-validation is frequently used. In this process, the data is divided into k datasets. $k - 1$ of them are used to train the model and the remaining one is used for validation. This process is repeated k times, by employing each of the possible datasets for validation. The final result is given by the arithmetic mean of the k results (Fig. 1.4). Leave-one-out Cross-Validation (LOOCV) is the special case

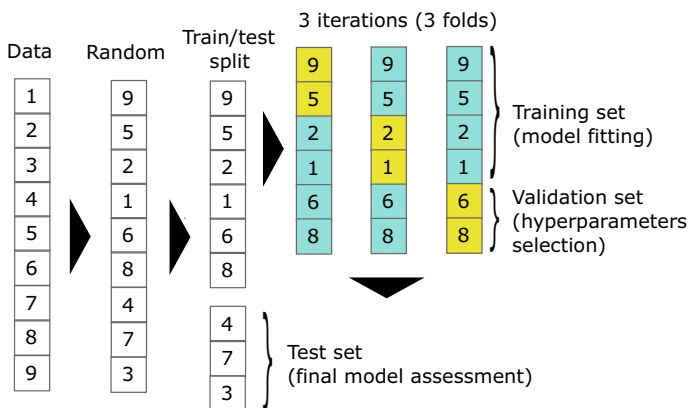


Fig. 1.4 Training and test sets: k-fold generation of training and validation sets from data. Number of data: 9, data for training and model selection: 6, data for final validation test (test set): 3, number of folds for model selection: 3, data in each fold: 2, number of models: 3 ($k = 3$). Sometimes, the validation test is also considered as test set. The 10-fold cross-validation is a common choice

where the number of folds is the same as the number of samples, so the test set has only one element. While LOOCV is expensive in general (Meijer and Goeman 2013), for the linear case it is very efficient because all the errors are obtained simultaneously with a single fit through the so-called hat matrix (Angelov and Stoimenova 2017).

1.2.2 Overview of Classical Machine Learning Procedures Used in CAE

The schemes we present in this section are basic ingredients of many ML algorithms.

1.2.2.1 Simple Regression Algorithms

The simplest ML approach is much older than the ML discipline: linear and nonlinear regression. In the former case, the purpose is to compute the weights \mathbf{w} and the offset b of the linear model $\tilde{y} \equiv f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b$, where \mathbf{x} is the vector of features. The parameters \mathbf{w} , b are obtained through the minimization of the cost function (MSE: Mean Squared Error)

$$C(\mathbf{x}_i; \{\mathbf{w}, b\}) = \frac{1}{n} \sum_{i=1}^n \mathcal{L}_i := \frac{1}{n} \sum_{i=1}^n [f(\mathbf{x}_i; \{\mathbf{w}, b\}) - y_i]^2 \quad (1.2)$$

with respect to them, which in this case is the average of the *loss function* $\mathcal{L}_i = (\tilde{y}_i - y_i)^2$, where the y_i are the known values, $\tilde{y}_i = f(\mathbf{x}_i; \{\mathbf{w}, b\})$ are the predictions, and the subindex i refers to sample i , so \mathbf{x}_i is the vector of features of that sample. Of course, in linear regression, the parameters are obtained simply by solving the linear system of equations resulting from the quadratic optimization problem. Other regression algorithms are similar, as for example spline, B-spline regressions, or P-spline (penalized B-splines) regressions, used in nonlinear mechanics (Crespo et al. 2017; Latorre and Montáns 2017) or used to perform efficiently an inverse of functions which does not have an analytical solution (Benítez and Montáns 2018; Eubank 1999; Eilers and Marx 1996). In all these cases, smoothing techniques are fundamental to avoid overfitting.

While it is natural to state the regression problem as a minimization of the cost function, it may be also formulated in terms of the likelihood function \mathcal{L} . Given some training data (y_i, \mathbf{x}_i) (with labels y_i for data \mathbf{x}_i), we seek the parameters \mathbf{w} (for simplicity we now include b in the set \mathbf{w}) that minimize the cost function (e.g., MSE); or equivalently we seek the set \mathbf{w} which maximizes the likelihood $\mathcal{L}(\mathbf{w}|(y, \mathbf{x})) = p(y|\mathbf{x}; \mathbf{w})$ for those parameters \mathbf{w} to give the probability representation for the training data, which is the same as the probability of finding the data (y, \mathbf{x}) given the distribution by \mathbf{w} . The likelihood is the “probability” by which a distribution (characterized by \mathbf{w}) represents all given data, whereas the probability is that of finding data if the distribution is known. Assuming data to be identically distributed and independent such

that $p(y_1, y_2, \dots, y_n | \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n; \mathbf{w}) = p(y_1 | \mathbf{x}_1; \mathbf{w}) p(y_2 | \mathbf{x}_2; \mathbf{w}) \dots p(y_n | \mathbf{x}_n; \mathbf{w})$, the likelihood is

$$\mathcal{L}(\mathbf{w} | (y_i, \mathbf{x}_i), i = 1, \dots, n) = \prod_{i=1}^n p(y_i | \mathbf{x}_i; \mathbf{w}) \quad (1.3)$$

or

$$\log \mathcal{L}(\mathbf{w} | (y_i, \mathbf{x}_i), i = 1, \dots, n) = \sum_{i=1}^n \log p(y_i | \mathbf{x}_i; \mathbf{w}) \quad (1.4)$$

Choosing the linear regression $\tilde{y} = \mathbf{w}^T \mathbf{x}$ (including b and 1 respectively in \mathbf{w} and \mathbf{x}), and a normal distribution of the prediction, obtained by assuming a zero-centered normal distribution of the error

$$p(y | \mathbf{x}; \mathbf{w}, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(\mathbf{w}^T \mathbf{x} - y)^2}{2\sigma^2}\right) \quad (1.5)$$

it is immediate to verify that the maximization of the log-likelihood in Eq. (1.4) is equivalent to minimizing the MSE in Eq. (1.2) (regardless of the value of the variance σ^2).

A very typical regression used in ML is logistic regression (Kleinbaum et al. 2002; Hosmer Jr et al. 2013), for example to obtain pass/fail (1/0) predictions. In this case, a smooth predictor output $y \in [0, 1]$ can be interpreted as a probability $p(y = 1 | \mathbf{x}) := p(\mathbf{x})$. The Bernoulli distribution (which gives p for $y_i = 1$ and $(1 - p)$ for $y_i = 0$), or equivalently

$$P(y = y_i) \equiv p^{y_i} (1 - p)^{(1 - y_i)} \quad (1.6)$$

describes this case for a given \mathbf{x}_i , as it is immediate to check. The linear regression is assigned to the *logit* function to convert the $(-\infty, \infty)$ range into the desired probabilistic $[0, 1]$ range

$$\text{logit}(p(\mathbf{x})) = \log\left(\frac{p(\mathbf{x})}{1 - p(\mathbf{x})}\right) = \mathbf{w}^T \mathbf{x} \quad (1.7)$$

The *logit* function is the logarithm of the ratio between the odds of $y = 1$ (which are p) and $y = 0$ (which are $(1 - p)$). The probability $p(\mathbf{x})$ may be factored out from Eq. (1.7) as

$$p(\mathbf{x}) = \frac{1}{1 + \exp(-\mathbf{w}^T \mathbf{x})} \quad (1.8)$$

which is known as the *sigmoid* function. Neural Networks frequently use logistic regression with the sigmoid model function where the parameters are obtained through the minimization of the proper cost function, or through the maximization

of the likelihood. In this latter case, the likelihood of the probability distribution in Eq. (1.6) is

$$\mathcal{L}(p(\mathbf{x})|(y_i, \mathbf{x}_i), i = 1, \dots, n) = \prod_{i=1}^n p(\mathbf{x}_i)^{y_i} [1 - p(\mathbf{x}_i)]^{(1-y_i)} \quad (1.9)$$

where y_i are the labels (with value 1 or 0) and $p(\mathbf{x}_i)$ are their (sigmoid-based) probabilistic predicted values given by Eq. (1.8) for the training data, which are a function of the parameters \mathbf{w} . The maximization of the log-likelihood of Eq. (1.9) for the model parameters gives the same solution as the minimization of the cross-entropy

$$\arg \min_{\mathbf{w}} H(\mathbf{w}) = - \sum_{i=1}^n \left[\frac{y_i}{n} \log p(\mathbf{x}_i; \mathbf{w}) + \frac{(1 - y_i)}{n} \log(1 - p(\mathbf{x}_i; \mathbf{w})) \right] \quad (1.10)$$

Another type of regression often used in ML is Kernel Regression. A Kernel is a positive-definite, typically non-local, symmetric weighting function $K(\mathbf{x}_i, \mathbf{x}) = K(\mathbf{x}, \mathbf{x}_i)$, centered in the attribute, with unit integral. The idea is similar to the use of shape functions in finite element formulations. For example, the Gaussian Kernel is

$$K(\mathbf{x}_i, \mathbf{x}) \equiv K_i(\mathbf{x}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left[-\frac{1}{2} \left(\frac{|\mathbf{x} - \mathbf{x}_i|}{\sigma} \right)^2 \right] \quad (1.11)$$

where σ is the bandwidth or smoothing parameter (deviation), and the weight for sample i is $w_i(\mathbf{x}) = K_i(\mathbf{x}) / \sum_{j=1}^n K_j(\mathbf{x})$. The predictor, using the weights from the kernel, is $f(\mathbf{x}) = \sum_{i=1}^n w_i(\mathbf{x}) y_i$ (although kernels may be used also for the labels). The cost function to determine σ^2 or other kernel parameters may be

$$\int f^2(\mathbf{x}) d\Omega - 2 \frac{1}{n} \sum_{i=1}^n f^{(i)}(\mathbf{x}_i) \longrightarrow \min \quad (1.12)$$

where the last summation term is the LOOCV, which excludes sample i from the set of predictions (recall that there are n different $f^{(i)}$ functions). Equation (1.12) focuses in essence on the minimum squared error for the solution. As explained below, kernels are also employed in Support Vector Machines to deal with nonlinearity and in dimensionality reduction of nonlinear problems to reduce the space.

1.2.2.2 Naïve Bayes

Naïve Bayes (NB) schemes are frequently used for classification (spam e-mail filtering, seismic vulnerability, etc.), and may be Multinomial NB or Gaussian NB. In both cases the probability theory is employed.

NB procedures operate as follows. From the training data, the prior probabilities for the different classes are computed, e.g., vulnerable or safe, $p(V)$ and $p(S)$, respectively, in our seismic vulnerability example. Then, for each feature, the probabilities are computed within each class, e.g., the probability that a vulnerable (or safe) structure is made of steel, $p(\text{steel}|V)$ (or $p(\text{steel}|S)$). Finally, given a sample outside the training set, the classification is obtained from the largest probability considering the class and the features present in the sample, e.g., $p(V)p(\text{steel}|V)p(\dots|V)\dots$ or $p(S)p(\text{steel}|S)p(\dots|S)\dots$, and so on. Gaussian NB are applied when the features have continuous (Gaussian) distributions, as for example the height of a building in the seismic vulnerability example. In this case the feature conditioned probabilities $p(\cdot|V)$ are obtained from the respective normal distributions. Logarithms of the probabilities are frequently used to avoid underflows.

1.2.2.3 Decision Trees (DT)

Decision trees are nonparametric. The simplest and well-known decision tree generator is the Iterative Dichotomiser 3 (ID3) algorithm, a greedy strategy. The scheme is used in the “guess who?” game and is also in essence the idea behind the root-finding bisection method or the Cuthill–McKee renumbering algorithm. The objective is, starting from one root node, to select at each step the feature and condition from the data that maximizes the Information Gain G (maximizes the benefit of the split), resulting in two (or more) subsequent leaf nodes. For example, in a group of people, the first optimal condition (maximizing the benefit of the split) is typically if the person is male or female, resulting in the male leaf and in the female leaf, each with 50% of the population, and so on. In seismic vulnerability, it could be if the structure is made of masonry, steel, wood, or Reinforced Concrete (RC). The gain G is the difference between the information entropies before ($H(S)$, parent entropy) and after the split given by the feature at hand j . Let us denote by $x_{j(i)}$ the feature j of sample i , by \mathbf{x}_i the array of features of sample i , and by x_j the different features (we omit the sample index if no confusion is possible). If $H(S|x_j)$ are the children entropies after the split by feature x_j , the Gain is

$$G(S, x_j) = H(S) - H(S|x_j) = H(S) - \sum_{s_i = \{\mathbf{x}_i, y_i\} \in S_j} p_j H(S_j) \quad (1.13)$$

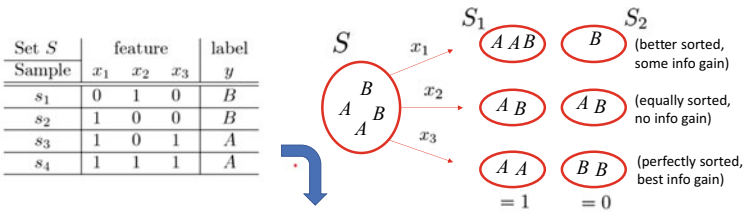
where the S_j are the subsets of S as a consequence of the split using feature (or attribute) x_j , p_j is the subset probability (number of samples s_i in subset S_j divided by the number of samples in the complete set S), and

$$H(S) = - \sum_{j=1}^l p(y_j) \log p(y_j) \quad (1.14)$$

where $H(S)$ is the information entropy of set S for the possible labels $y_j, j = 1, \dots, l$, so Eq. (1.13) results in

$$G(S, x_j) = - \sum_{i=1}^l p(y_i) \log p(y_i) + \sum_{s_i \in S_j} p_j \sum_{i=1}^l p(y_i|x_j) \log p(y_i|x_j) \quad (1.15)$$

The gain G is computed for each feature x_j (e.g., windows, structure type, building age, and soil type). The feature that maximizes the Gain is the one selected to generate the next level of leaves. The decision tree building process ends when the entropy reaches zero (the samples are perfectly classified). Figure 1.5 shows a simple example with four samples s_i in the dataset, each with three features x_j of two possible values (0 and 1), and one label y of two possible values (A and B). The best of the three features is selected as that one which provides the most information gain. It is seen that feature 1 produces some information gain because after the split using this feature, the samples are better classified according to the label. Feature 2 gives no



Set S or sub-sets S_1, S_2	Feature	Samples label A	Samples label B	n_A	n_B	$n_A + n_B$	Parent or child entropies $H(S)$	p	Inform. Gain G
Full set S	(any)	s_3, s_4	s_1, s_2	2	2	4	$-\frac{2}{4} \log_2 \frac{2}{4} - \frac{2}{4} \log_2 \frac{2}{4} = 1$		N/A
S_1 for x_1	$x_1 = 1$	s_3, s_4	s_2	2	1	3	$-\frac{2}{3} \log_2 \frac{2}{3} - \frac{1}{3} \log_2 \frac{1}{3} = 0.92$	$\frac{3}{4}$	$1 - \frac{3}{4} \cdot 0.92 - \frac{1}{4} \cdot 0$
S_2 for x_1	$x_1 = 0$	\emptyset	s_1	0	1	1	$-0 - \frac{1}{1} \log_2 1 = 0$	$\frac{1}{4}$	$= 0.31$ (some)
S_1 for x_2	$x_2 = 1$	s_4	s_1	1	1	2	$-\frac{1}{2} \log_2 \frac{1}{2} - \frac{1}{2} \log_2 \frac{1}{2} = 1$	$\frac{2}{4}$	$1 - \frac{2}{4} \cdot 1 - \frac{2}{4} \cdot 1$
S_2 for x_2	$x_2 = 0$	s_3	s_2	1	1	2	$-\frac{1}{2} \log_2 \frac{1}{2} - \frac{1}{2} \log_2 \frac{1}{2} = 1$	$\frac{2}{4}$	$= 0$ (no gain)
S_1 for x_3	$x_3 = 1$	s_3, s_4	\emptyset	2	0	2	$-\frac{2}{2} \log_2 \frac{2}{2} - 0 = 0$	$\frac{2}{4}$	$1 - \frac{2}{4} \cdot 0 - \frac{2}{4} \cdot 0$
S_2 for x_3	$x_3 = 0$	\emptyset	s_1, s_2	0	2	2	$-0 - \frac{2}{2} \log_2 \frac{2}{2} = 0$	$\frac{2}{4}$	$= 1$ (best gain)

Fig. 1.5 Example of determination of the feature with most information gain. If we choose feature x_1 for sorting, we find two subsets, $S_1 = \{s_i \text{ such that } x_1 = 1\} = \{s_2, s_3, s_4\}$ and $S_2 = \{s_i \text{ s.t. } x_1 = 0\} = \{s_1\}$. In S_1 there are two elements (s_3, s_4) with label A , and one element (s_2) with label B , so the probabilities are $2/3$ for label A and $1/3$ for label B . In S_2 the only element (s_1) has label B , so the probabilities are $0/1$ for label A and $1/1$ for label B . The entropy of subset S_1 is $H(S_1) = -\frac{2}{3} \log_2 \frac{2}{3} - \frac{1}{3} \log_2 \frac{1}{3} = 0.92$. The entropy of subset S_2 is $H(S_2) = -0 - \frac{1}{1} \log_2 \frac{1}{1} = 0$. In a similar form, since there are a total of 4 samples, two with label A and two with label B , the parent entropy is $H(S) = -\frac{2}{4} \log_2 \frac{2}{4} - \frac{2}{4} \log_2 \frac{2}{4} = 1$. Then, the information gain is $G = H(S) - p(S_1)H(S_1) - p(S_2)H(S_2) = 1 - \frac{3}{4} \cdot 0.92 - \frac{1}{4} \cdot 0 = 0.31$, where $p(S_i)$ is the probability of a sample being in subset S_i , i.e. $3/4$ for S_1 and $1/4$ for S_2 . Repeating the computations for the other two features, it is seen that feature x_3 is the one that has the best information gain. Indeed the information gain is $G = 1$ because it fully separates the samples according to the labels

gain because it is useless to distinguish the samples according to the label (it is in 50% each), and feature 3 is the best one because it fully classifies the samples according to the label (A is equivalent to $x_3 = 1$, and B is equivalent to $x_3 = 0$). As for the Cuthill–McKee renumbering algorithm, there is no proof of reaching the optimum.

While DT are typically used for classification, there are regression trees in which the output is a real number. Other decision tree algorithms are the C4.5 (using continuous attributes), Classification And Regression Tree (CART), and Multivariate Adaptive Regression Spline (MARS) schemes.

1.2.2.4 Support Vector Machines (SVM), k-Means, and k-Nearest Neighbors (kNN)

The Support Vector Machine (SVM) is a technique which tries to find the optimal hyperplane separating groups of samples for clustering (unsupervised) or classification (supervised). Consider the function $z(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b$ for classification. The model is

$$y \equiv f(\mathbf{x}) = \text{sign}(z(\mathbf{x})) = \text{sign}(\mathbf{w}^T \mathbf{x} + b) \quad (1.16)$$

where the parameters $\{\mathbf{w}, b\}$ are obtained by minimizing $\frac{1}{2}|\mathbf{w}|^2$ (or equivalently $|\mathbf{w}|^2$ or $|\mathbf{w}|$) subject to $y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 \quad \forall i$ such that the decision boundary $f(\mathbf{x}) = 0$ given by the hyperplane has maximum distance to the groups of the samples; see Fig. 1.6. The minimization problem (in *primal form*) using Lagrange multipliers α_i is

$$\text{find } \arg \min_{\mathbf{w}, b} \left[\frac{1}{2}|\mathbf{w}|^2 + \sum_{i=1}^n \alpha_i (1 - y_i(\mathbf{w}^T \mathbf{x}_i + b)) \right] \text{ with } \alpha_i \geq 0 \quad (1.17)$$

or in penalty form

$$\text{find } \arg \min_{\mathbf{w}, b} \left[\frac{1}{2}|\mathbf{w}|^2 + C \sum_{i=1}^n \max [0, 1 - y_i(\mathbf{w}^T \mathbf{x}_i + b)] \right] \quad (1.18)$$

A measure of certainty for sample i is based on its proximity to the boundary; i.e. $(\mathbf{w}^T \mathbf{x}_i + b)/|\mathbf{w}|$ (the larger the distance to the boundary, the more certain the classification of the sample). Of course, SVMs may be used for multiclass classification, e.g., using the One-to-Rest approach (employing k SVMs to classify k classes) or the One-to-One approach (employing $\frac{1}{2}k(k-1)$ SVMs to classify k classes); see Fig. 1.6.

Taking the derivative of the Lagrangian in squared brackets in Eq. (1.17) with respect to \mathbf{w} and b , we get that at the minimum

$$\mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i \quad \text{and} \quad \sum_{i=1}^n \alpha_i y_i = 0 \quad (1.19)$$

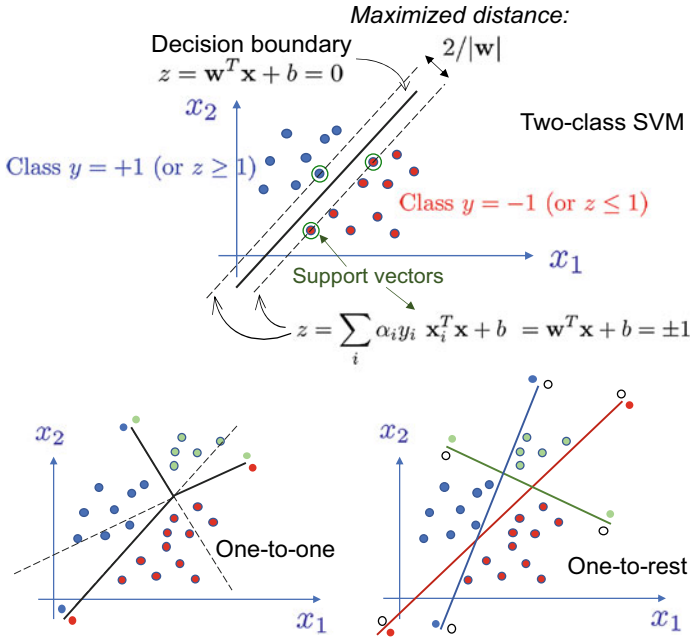


Fig. 1.6 Two-class SVM decision boundary and one-to-one and one-to-rest SVM multiclass classification

and substituting it in the primal form given in Eq. (1.17), the minimization problem may be written in its dual form

$$\text{find } \arg \max_{\alpha_i \geq 0} \left[\sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{j,k=1}^n \alpha_j \alpha_k y_j y_k (\mathbf{x}_j^T \mathbf{x}_k) \right] \quad \text{with } \sum_{i=1}^n \alpha_i y_i = 0 \quad (1.20)$$

and with $b = y_j - \mathbf{w}^T \mathbf{x}_j$ being \mathbf{x}_j any active (support) vector, with $\alpha_j > 0$. Then, $z = \mathbf{w}^T \mathbf{x} + b$ is $z = \sum_i \alpha_i y_i \mathbf{x}_i^T \mathbf{x} + b$. Instead of searching for the weights $w_i, i = 1, \dots, N$ (N is the number of features of each sample), we search for the coefficients $\alpha_i, i = 1, \dots, n$ (n is the number of samples).

Nonlinear separable cases may be addressed through different techniques as using positive slack variables $\xi_i \geq 0$ or kernels. When using slack variables (*Soft Margin SVM*), for each sample i we write $y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i$ and we apply a L1 (LASSO) regularization by minimizing $\frac{1}{2} |\mathbf{w}|^2 + C \sum_i \xi_i$ subject to the constraints $y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i$ and $\xi_i \geq 0$, where C is the penalization parameter. In this case, the only change in the dual formulation is the constraint for the Lagrange multipliers: $C \geq \alpha_i \geq 0$, as it can be easily verified.

When using kernels, the kernel trick is typically employed. The idea behind the use of kernels is that if data is linearly non-separable in the features space, it may be

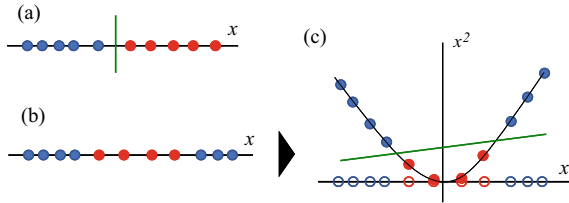


Fig. 1.7 Use of higher dimensions to obtain linearly separable data. **a** Data is linearly separable in 1D. **b** Data is not linearly separable in 1D. **c** Using two dimensions with mapping $\phi = [x, x^2]^T$, data becomes linearly separable in the augmented space

separable in a larger space; see, for example, Fig. 1.7. This technique uses the dual form of the SVM optimization problem. Using the dual form

$$|\mathbf{w}|^2 = \mathbf{w}^T \mathbf{w} = \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j (\mathbf{x}_i^T \mathbf{x}_j) \text{ and } \mathbf{w}^T \mathbf{x} = \sum_{i=1}^n \alpha_i y_i (\mathbf{x}_i^T \mathbf{x}) \quad (1.21)$$

the equations only involve inner products of feature vectors of the type $(\mathbf{x}_i^T \mathbf{x}_j)$, ideal for using a kernel trick. For example, the case shown in Fig. 1.8 is not linearly separable in the original features space, but using the mapping $\phi(\mathbf{x}) := [x_1^2, x_2^2, \sqrt{2}x_1x_2]^T$ to an augmented space, we find that the samples are linearly separable in this space. Then, for performing the linear separation in the transformed space, we have to compute z in that transformed space (Representer Theorem, Schölkopf et al. 2001)

$$z = \sum_{i=1}^n \alpha_i y_i [\phi(\mathbf{x}_i)^T \phi(\mathbf{x})] + b \quad (1.22)$$

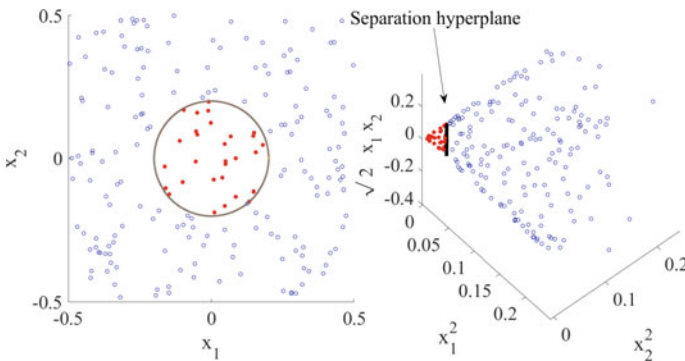


Fig. 1.8 Linearly non-separable samples (left). Linear separation in a transformed higher dimensional space (right)

to substitute the inner products in the original space by inner products in the transformed space. These operations (transformations plus inner products in the high-dimensional space) can be expensive (in complex cases we need to add many dimensions). However, in our example we note that

$$K(\mathbf{a}, \mathbf{b}) := \boldsymbol{\phi}(\mathbf{a})^T \boldsymbol{\phi}(\mathbf{b}) = [a_1^2 \ a_2^2 \ \sqrt{2}a_1a_2] \begin{bmatrix} b_1^2 \\ b_2^2 \\ \sqrt{2}b_1b_2 \end{bmatrix} = \left([a_1 \ a_2] \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \right)^2 = (\mathbf{a}^T \mathbf{b})^2 \quad (1.23)$$

so it is not necessary to use the transformed space because the inner product can be equally calculated in both spaces. Indeed note that, remarkably, we even do not need to know explicitly $\boldsymbol{\phi}(\mathbf{x})$, because the kernel $K(\mathbf{a}, \mathbf{b}) = (\mathbf{a}^T \mathbf{b})^2$ is fully written in the original space and we never need $\boldsymbol{\phi}(\mathbf{x})$. Then we just solve

$$\text{find } \arg \max_{\alpha_i \geq 0} \left[\sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{j,k=1}^n \alpha_j \alpha_k y_j y_k K(\mathbf{x}_j, \mathbf{x}_k) \right] \text{ with } \sum_{i=1}^n \alpha_i y_i = 0 \quad (1.24)$$

Examples of kernel functions are the polynomial $K(\mathbf{a}, \mathbf{b}) = (\mathbf{a}^T \mathbf{b} + 1)^d$, where d is the degree, and the Gaussian radial basis function (which may be interpreted as a polynomial form with infinite terms)

$$K(\mathbf{a}, \mathbf{b}) = \exp(-\gamma |\mathbf{a} - \mathbf{b}|) \quad (1.25)$$

where $\gamma = 1/(2\sigma^2) > 0$.

Considering clustering, another algorithm similar in nature to SVM is used and is called the k -means (unsupervised technique creating k clusters). The idea here is to employ a distance measurement in order to determine the optimal center of clusters and the optimal decision boundaries between clusters. The case $k = 1$ is essentially the Voronoi diagram. Another simple approach is the k -Nearest Neighbors (k -NN), a supervised technique also employed in classification. This technique uses the labels of the k -nearest neighbors to predict the label of the target points (e.g., by some weighting method).

1.2.2.5 Dimensionality Reduction Algorithms

When problems have too many features (or data, measurements), dimensionality reduction techniques are employed to reduce the number of attributes and gain insight into the most meaningful ones. These are typically employed not only in pattern recognition and image processing (e.g., identification or compression) but also to determine which features, data, or variables are most relevant for the learning purpose. In essence, the algorithms are similar in nature to determining the principal modes in a dynamic response, because with that information, relevant mechanical

properties (mass distribution, stiffness, damping), and the overall response may be obtained. In ML, classical approaches are given by Principal Component Analysis (PCA) based on Pearson’s Correlation Matrix (Abdi and Williams 2010; Bro and Smilde 2014), Singular Value Decomposition (SVD), Proper Orthogonal Decomposition (POD) (Berkooz et al. 1993), Linear (Fisher’s) Discriminant Analysis (LDA) (Balakrishnama and Ganapathiraju 1998; Fisher 1936), Kernel (Nonlinear) Principal Component Analysis (kPCA), Hofmann et al. (2008), Alvarez et al. (2012), Local Linear Embedding (LLE) (Roweis and Saul 2000; Hou et al. 2009), Manifold Learning (used also in constitutive modeling) (Cayton 2005; Bengio et al. 2013; Turaga et al. 2020), Uniform Manifold Approximation and Projection (UMAP) (McInnes et al. 2018), and autoencoders (Bank et al. 2020; Zhuang et al. 2021; Xu and Duraisamy 2020; Bukka et al. 2020; Simpson et al. 2021). Often, these approaches are also used in clustering.

LLE is one of the simplest nonlinear dimension reduction processes. The idea is to identify a global space with smaller dimension that reproduces the proximity of data in the higher dimensional space; it is a k -NN approach. First, we determine the weights w_{ij} , such that $\sum w_{ij} = 1$, which minimize the error

$$\text{Error}(\mathbf{w}) = \sum_{i=1}^n \left(\mathbf{x}_i - \sum_{j=1}^k w_{ij} \mathbf{x}_j \right)^2 \quad (1.26)$$

in the representation of a point from the local space given by the k -nearest points (k is a user-prescribed hyperparameter), so

$$\mathbf{w} = \arg \min [\text{Error}(\mathbf{w})] \quad (1.27)$$

Then, we search for the images \mathbf{y}_i of \mathbf{x}_i in the lower dimensional space, simply by considering that the computed w_{ij} reflect the geometric properties of the local manifold and are invariant to translations and rotations. Given w_{ij} , we now look for the lower dimensional coordinates \mathbf{y}_i that minimize the cost function

$$C(\mathbf{y}_i, i = 1, \dots, n) = \sum_{i=1}^n \left(\mathbf{y}_i - \sum_{j=1}^k w_{ij} \mathbf{y}_j \right)^2 \quad (1.28)$$

Isometric Mapping (ISOMAP) techniques are similar, but use geodesic k -node-to- k -node distances (computed by Dijkstra’s 1959 or the Floyd–Warshall 1962 algorithms to find the shortest path between nodes) and look for preserving them in the reduced space. Another similar technique is the Laplacian eigenmaps scheme (Belkin and Niyogi 2003), based on the non-singular lowest eigenvectors of the Graph Laplacian $\mathbf{L} = \mathbf{d} - \mathbf{w}$, where $d_{ii} = \sum_j w_{ij}$ gives the diagonal degree matrix and w_{ij} are the edge weights, computed for example using the Gaussian kernel $w_{ij} = K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-|\mathbf{x}_i - \mathbf{x}_j|^2 / (2\sigma^2))$. Within the same k -neighbors family, yet more complex and advanced, are Topological Data Analysis (TDA) techniques. A valuable overview may be found in Chazal and Michel (2021); see also the references therein.

For the case of PCA, it is typical to use the covariance matrix

$$S_{jk} = \frac{1}{n} \sum_{i=1}^n (x_{j(i)} - \bar{x}_j)(x_{k(i)} - \bar{x}_k) \tag{1.29}$$

where the overbar denotes the mean value of the feature, and $x_{j(i)}$ is feature j of sample i . The eigenvectors and eigenvalues of the covariance matrix are the principal components (directions/values of maximum significance/relevance), and the number of them selected as sufficient is determined by the variance ratios; see Fig. 1.9(a). PCA is a linear unsupervised technique. The typical implementation uses mean-corrected samples, as in kPCA, so in such case $S_{jk} = \frac{1}{n} \sum_{i=1}^n x_{j(i)}x_{k(i)}$, or in matrix notation $\mathbf{S} = \frac{1}{n}\mathbf{X}\mathbf{X}^T$. kPCA (Schölkopf et al. 1997) is PCA using kernels (such as polynomials, the hyperbolic tangent, or the Radial Basis Function (RBF)) to address the nonlinearity by expanding the space. For example, using the RBF, we construct the kernel matrix K_{ij} , for which the components are obtained from the samples i, j as $K_{ij} = \exp(-\gamma|\mathbf{x}_i - \mathbf{x}_j|^2)$. The RBF is then centered in the transformed space by (note that being centered in the original features space does not mean that the features are also automatically centered in the transformed space, hence the need for this operation)

$$\bar{\mathbf{K}} = \mathbf{K} - \frac{1}{n}\mathbf{1}\mathbf{K} - \frac{1}{n}\mathbf{K}\mathbf{1} + \frac{1}{n^2}\mathbf{1}\mathbf{K}\mathbf{1} \tag{1.30}$$

where $\mathbf{1}$ is a $n \times n$ matrix of the unit entry “1”. Then $\bar{\mathbf{K}}(\mathbf{x}_i, \mathbf{x}_j) = \bar{\phi}(\mathbf{x}_i)^T \bar{\phi}(\mathbf{x}_j)$ with the centered $\bar{\phi}(\mathbf{x}_i) = \phi(\mathbf{x}_i) - 1/n \sum_{r=1}^n \phi(\mathbf{x}_r)$. The larger eigenvalues are the

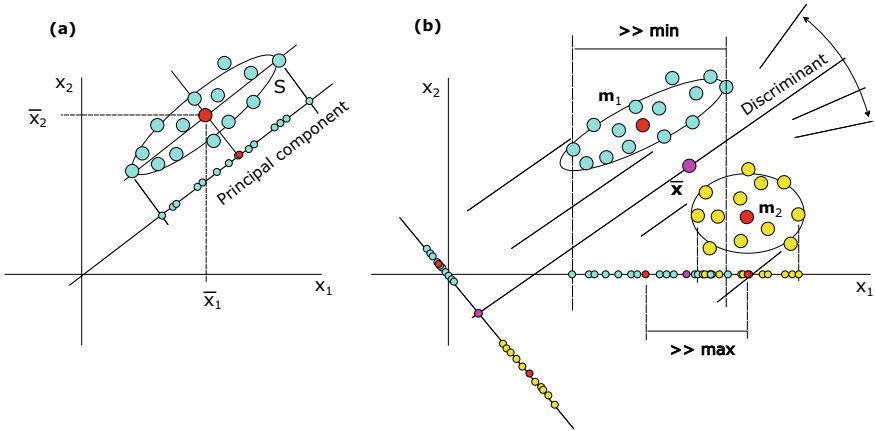


Fig. 1.9 **a** Principal Component Analysis. The principal components are those with largest variations (largest eigenvalues of the variances matrix). **b** Linear Discriminant Analysis to separate clusters. It is seen that feature x_1 is not a good choice to determine if a sample belongs to a given cluster, but there is a features combination (a line) which gives the best discrimination between clusters. That combination maximizes the distance between means of clusters whereas minimizes the dispersion of samples within the clusters

principal components in the transformed space, and the corresponding eigenvectors are the samples already projected onto the principal axes.

In contrast to PCA, the purpose of LDA is typically to improve separability of known classes (a supervised technique), and hence maximize information in this sense: maximizing the distance between the mean values of the classes and, within each class, minimizing the variation. It does so through the eigenvalues of the normalized between-classes scatter matrix $\mathbf{S}_w^{-1}\mathbf{S}_b$ (the *between-variances* by the *within-variances*) where

$$\mathbf{S}_w = \sum_{i=1}^{n\text{-classes}} \sum_{\text{within-class-}i}^{n_i} (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^T \quad (1.31)$$

$$\mathbf{S}_b = \sum_{\text{Class } i=1}^{n\text{-classes}} n_i (\mathbf{m}_i - \bar{\mathbf{x}})(\mathbf{m}_i - \bar{\mathbf{x}})^T \quad (1.32)$$

and $\bar{\mathbf{x}}$ is the overall mean vector of the features \mathbf{x} and \mathbf{m}_i is the mean vector of those within-class i . If $\bar{\mathbf{x}} = \mathbf{m}_i$ the class is not separable from the selected features. Frequently used nonlinear extensions of LDA are the Quadratic Discriminant Analysis (QDA) (Tharwat 2016; Ghosh et al. 2021), Flexible Discriminant Analysis (FDA) (Hastie et al. 1994), and Regularized Discriminant Analysis (RDA) (Friedman 1989).

Proper Orthogonal Decompositions (POD) are frequently motivated in PCA and are often used in turbulence and in reducing dynamical systems. It is a technique also similar to classical modal decomposition. The idea is to decompose the time-dependent solution as

$$\mathbf{u}(\mathbf{x}, t) = \sum_{p=1}^P a_p(t) \boldsymbol{\varphi}_p(\mathbf{x}) \quad (1.33)$$

and compute the Proper Orthogonal Modes (POMs) $\boldsymbol{\varphi}_p(\mathbf{x})$ that maximize the energy representation (L2-norm). In essence, we are looking for the set of “discrete functions” $\boldsymbol{\varphi}_p(\mathbf{x})$ that best represent $\mathbf{u}(\mathbf{x}, t)$ with the lowest number of terms P . Since these are computed as discretized functions, several snapshots $\mathbf{u}(\mathbf{x}, t_i)$, $i = 1, \dots, n$ are grabbed in the discretized domain, i.e.

$$\mathbf{U} = [\mathbf{u}(\mathbf{x}, t_1) \ \mathbf{u}(\mathbf{x}, t_2) \ \dots \ \mathbf{u}(\mathbf{x}, t_n)] = \begin{bmatrix} u_{11} & \dots & u_{1n} \\ \vdots & \ddots & \vdots \\ u_{m1} & \dots & u_{mn} \end{bmatrix} \quad (1.34)$$

Then, the POD vectors are the eigenvectors of the sample covariance matrix. If the snapshots are corrected to have zero mean value, the covariance matrix is

$$\mathbf{S} = \frac{1}{n} \mathbf{U} \mathbf{U}^T \quad (1.35)$$

The POMs may also be computed using the SVD of \mathbf{U} (the left singular vectors are the eigenvectors of $\mathbf{U}\mathbf{U}^T$) or auto-associative NNs (Autoencoder Neural Networks that replicate the input in the output but using a hidden layer of smaller dimension). To overcome the curse of dimensionality when using too many features (e.g., for parametric analyses), the POD idea is generalized in Proper Generalized Decomposition (PGD), by assuming approximations of the form

$$\mathbf{u}(x_1, x_2, \dots, x_d) = \sum_{i=1}^N \boldsymbol{\phi}_i^1(x_1) \circ \boldsymbol{\phi}_i^2(x_2) \circ \dots \circ \boldsymbol{\phi}_i^d(x_d) \quad (1.36)$$

where $\boldsymbol{\phi}_i^j(x_j)$ are the unknown vector functions (usually also discretized and computed iteratively, for example using a greedy algorithm), and “ \circ ” stands for the Hadamard or entry-wise product of vectors. Note that, in general we cannot use the separability $\Phi(x, y) \neq \phi(x)\psi(y)$ but PGDs look for the best $\phi_i(x)\psi_i(y)$ choices for the given problem such that we can say $\Phi(x, y) \simeq \sum_i \phi_i(x)\psi_i(y)$ in a sufficiently small number of addends (hence with a reduced complexity). The power of the idea is that for a large number n of features, determining functions of the type $\Phi(x_1, x_2, \dots, x_n)$ is virtually impossible, but determining products and additions of scalar functions is feasible.

The UMAP and t-SNE schemes are based on the concept of a generalized metric or distance between samples. A symmetric and normalized (between 0 and 1) metric is defined as

$$d_{ij}(\mathbf{x}_i, \mathbf{x}_j) = d_i^j(\mathbf{x}_i, \mathbf{x}_j) + d_j^i(\mathbf{x}_i, \mathbf{x}_j) - d_i^j(\mathbf{x}_i, \mathbf{x}_j)d_j^i(\mathbf{x}_i, \mathbf{x}_j) \quad (1.37)$$

where the unidirectional distance function is defined as

$$d_i^j(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\rho_{ij} - \rho_i^1}{\rho_i^k}\right) \quad (1.38)$$

where $\rho_{ij} = |\mathbf{x}_i - \mathbf{x}_j|$ and $\rho_i^k = |\mathbf{x}_i - \mathbf{x}_k|$, with k referring to the k -nearest neighbor (ρ_i^1 refers to the nearest neighbor to i). Here k is an important hyperparameter. Note that $d_i^j = 1$ if i, j are nearest neighbors, and $d_i^j \rightarrow 0$ for far away neighbors. We are looking for a new set of lower dimensional features \mathbf{z} to replace \mathbf{x} . The same generalized distance $d_{ij}(\mathbf{z}_i, \mathbf{z}_j)$ may be applied to the new features. To this end, through optimization techniques, like the steepest descent, we minimize the fuzzy set dissimilarity cross-entropy (or entropy difference) like the Kullback–Leibler (KL) divergence (Hershey and Olsen 2007; Van Erven and Harremos 2014), which measures the difference between the probability distributions $d_{ij}(\mathbf{x}_i, \mathbf{x}_j)$ and $d_{ij}(\mathbf{z}_i, \mathbf{z}_j)$, and their complementary values $[1 - d_{ij}(\mathbf{x}_i, \mathbf{x}_j)]$ and $[1 - d_{ij}(\mathbf{z}_i, \mathbf{z}_j)]$ (recall that $d \in (0, 1]$, so it is seen as a probability distribution)

$$KL(d(\mathbf{x}), d(\mathbf{z})) = \sum_{i,j=1}^n \left\{ d_{ij}(\mathbf{x}_i, \mathbf{x}_j) \ln \left[\frac{d_{ij}(\mathbf{x}_i, \mathbf{x}_j)}{d_{ij}(\mathbf{z}_i, \mathbf{z}_j)} \right] + [1 - d_{ij}(\mathbf{x}_i, \mathbf{x}_j)] \ln \left[\frac{1 - d_{ij}(\mathbf{x}_i, \mathbf{x}_j)}{1 - d_{ij}(\mathbf{z}_i, \mathbf{z}_j)} \right] \right\} \quad (1.39)$$

Note that the KL scheme is not symmetric with respect to the distributions. If distances in both spaces are equal for all the samples, $KL = 0$. In general, a lower dimensional space gives $KL \neq 0$, but with the dimension of \mathbf{z} fixed, the features (or combinations of features) that give a minimum KL considering all n samples represent the optimal selection.

Autoencoders are a type of neural network, discussed below, and can be interpreted as a nonlinear generalization of PCA. Indeed, an autoencoder with linear activation functions is equivalent to a SVD.

1.2.2.6 Genetic Algorithms

Genetic Algorithms (Mitchell 1998) in ML (or more generally evolutionary algorithms) are in essence very similar to those employed in optimization (Grefenstette 1993; De Jong 1988). They are metaheuristic algorithms which include the steps in natural evolution: (1) Initial population, (2) a fitness function, (3) a (nature-like) selection according to fitness, (4) crossover (the gene combination), (5) mutation (random alteration). After running many generations, convergence is expected to the superspecies. Feature selection and database reduction is a typical application (Vafaie and De Jong 1992). The variety of implementations is large and the implementations depend on the specific problem addressed (e.g., polymer design, Kim et al. 2021, and materials modeling, Paszkowicz 2009), but the essence and ingredients are similar.

1.2.2.7 Rosenblatt’s Perceptron, Adaline (Adaptive Linear Neuron) Model, and Feed Forward Neural Networks (FFNN)

Currently, the majority of ML algorithms employed in practice are some type or variation of Neural Networks. Deep Learning (DL) refers to NNs with many layers. While the NN theory was proposed some decades ago, efficient implementations facilitating the solution of real-world problems have been established only in the late 1980s and early 1990s. NNs are based on the ideas from McCulloch and Pitts (1943) describing a simple model for the work of neurons, and on Rosenblatt’s perceptron (Rosenblatt 1958); see Fig. 1.10. The Adaline model (Widrow and Hoff 1962) (Fig. 1.10) introduces the activation function to drive the learning process from the different samples, instead of the dichotomic outputs from the samples. This activation function is today one of the keystones of NNs. The logistic sigmoid function is frequently used. There are other alternatives such as the ReLU (Rectified Linear Unit; the Macaulay ramp function) or the hyperbolic tangent

$$\tanh(z) = \frac{\exp(z) - \exp(-z)}{\exp(z) + \exp(-z)}, \text{ with } z = \mathbf{w}^T \mathbf{x} + w_0 \quad (1.40)$$

NNs are made from many such artificial neurons, typically arranged in several layers, with each layer $l = 1, \dots, L$ containing many neurons. The output from the network

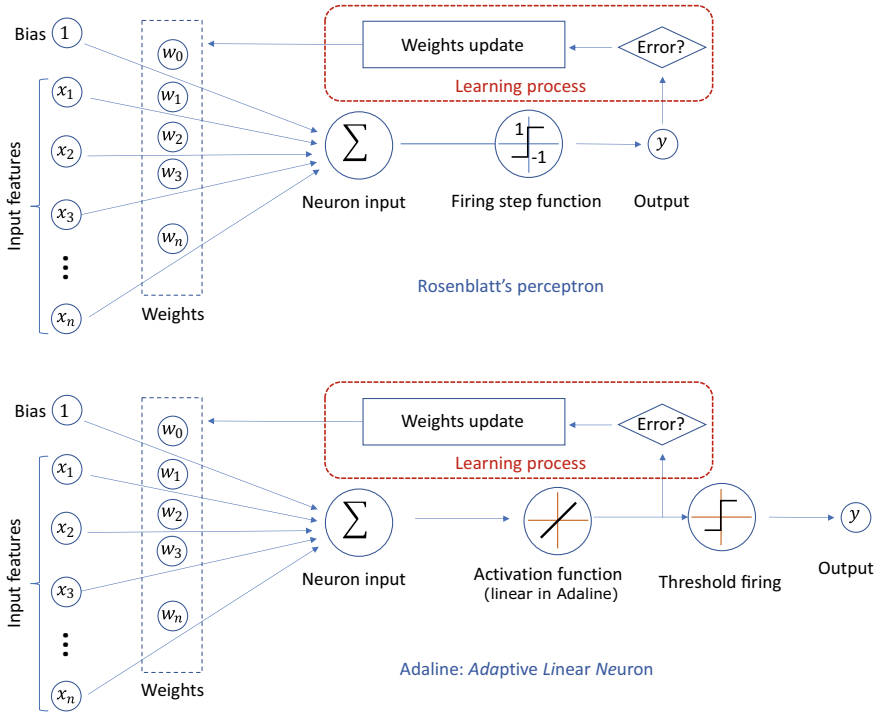


Fig. 1.10 Rosenblatt’s perceptron and Adaline (Adaptive Linear Neuron) model

is defined as a composition of functions

$$\mathbf{y} = \mathbf{f}^L(\mathbf{f}^{L-1}(\mathbf{f}^{L-2}(\dots \mathbf{f}^2(\mathbf{f}^1(\mathbf{z}^1)))))) \tag{1.41}$$

where the \mathbf{f}^l are the neuron functions of the layer (often also denoted by σ^l in the sigmoid case), typically arranged by groups in the form $\mathbf{f}^l(\mathbf{W}^l \mathbf{x}^l + \mathbf{b}_l)$, where \mathbf{W}^l is the matrix of weights, $\mathbf{z}^l := \mathbf{W}^l \mathbf{x}^l + \mathbf{b}_l$, $\mathbf{x}^l = \mathbf{y}^{l-1} = \mathbf{f}^{l-1}(\mathbf{z}^{l-1})$ are the neuron inputs and output of the previous layer (the features for the first function; $\mathbf{y}^0 \equiv \mathbf{x}$), and \mathbf{b}_l is the layer bias vector, which is often incorporated as a weight on a unit bias by writing $\mathbf{z}^l = \mathbf{W}^l \mathbf{x}^l$, so \mathbf{x}^l has also the index 0, and $x_0^l = 1$; see Fig. 1.11. The output may be also a vector $\mathbf{y} \equiv \mathbf{y}^L$. The purpose of the learning process is to learn the optimum values of \mathbf{W}^l , \mathbf{b}_l . The power of the NNs is that a simple architecture, with simple functions, may be capable of reproducing more complex functions. Indeed, Rosenblatt’s scheme discussed below may give any linear or nonlinear function. Of course, complex problems will require many neurons, layers, and data, but the overall structure of the method is almost unchanged.

The Feed Forward Neural Network (FFNN) with many layers, as shown in Fig. 1.11, is trained by optimization algorithms (typically modifications of the steepest descent) using the backpropagation algorithm, which consists in computing the

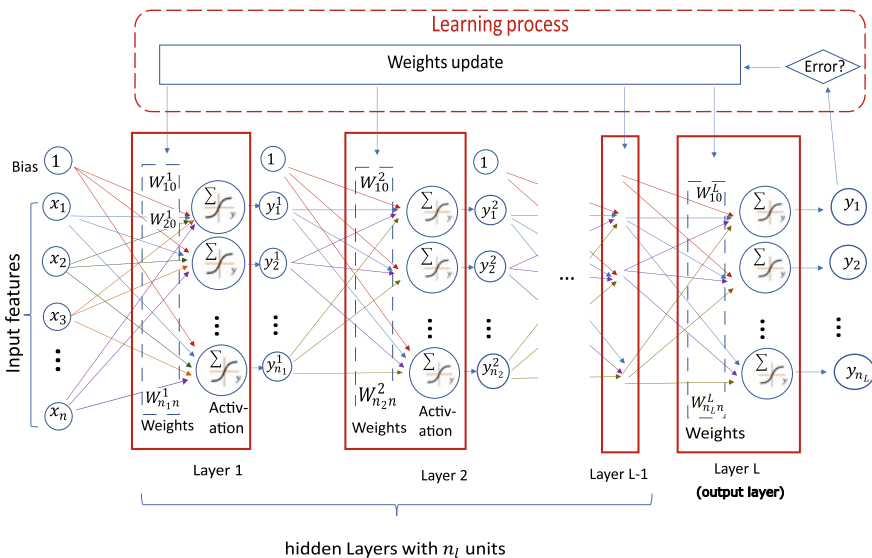


Fig. 1.11 Neural Network with $L - 1$ hidden layers and one (the L) output layer. Notation for weights is W_{oi}^l , where i is the input cell (zero refers to the bias unit), o is the output cell (the order is often reversed in the literature), and $l = 1, \dots, L$ are the layers

sensitivities using the chain rule from the output layer to the input layer, so for each layer, the information of the derivatives of the subsequent layers are known. For example, in Fig. 1.11, assume that the error is computed as $E = \frac{1}{2} (\mathbf{y} - \mathbf{y}^{\text{exp}})^T (\mathbf{y} - \mathbf{y}^{\text{exp}})$ (logistic errors are more common, but we consider herein this simpler case). Then, if α is the learning rate (a hyperparameter), the increment between epochs of the parameters is

$$\Delta W_{oi}^l = -\alpha (\mathbf{y} - \mathbf{y}^{\text{exp}})^T \frac{\partial \mathbf{y}}{\partial W_{oi}^l} \quad (1.42)$$

where $\partial \mathbf{y} / \partial W_{oi}^l$ is computed through the chain rule. Figure 1.12 shows a simple example with two layers and two neurons per layer; superindices refer to layer and subindices to neuron. For example, following the green path, we can compute

$$\frac{\partial y_2}{\partial W_{21}^2} = \left[\frac{\partial y_2}{\partial z_2^2} \right] \left\{ \frac{\partial z_2^2}{\partial W_{21}^2} \right\} \quad (1.43)$$

where $\partial y_2 / \partial z_2^2$ is the derivative of the selected activation function evaluated at the iterative value z_2^2 and $\partial z_2^2 / \partial W_{21}^2 = x_1^2$ is also the known iterative value. As an example of a deeper layer, consider the red line in Fig. 1.12

$$\frac{\partial y_1}{\partial W_{21}^1} = \left[\frac{\partial y_1}{\partial z_1^1} \right] \left[\frac{z_1^2}{\partial x_2^2} \frac{\partial x_2^2}{\partial z_2^1} \right] \left\{ \frac{\partial z_2^1}{\partial W_{21}^1} \right\} \quad (1.44)$$

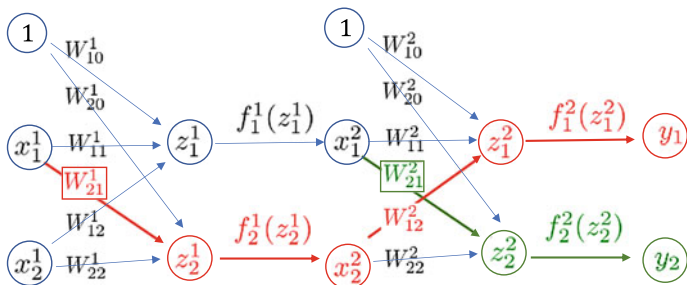


Fig. 1.12 Computation of the gradient through backpropagation. z_o^l is defined as $z_o^l = W_{oi}^l x_i^l$ (which includes the bias) and $f_o^l(z_o^l)$ is the activation function

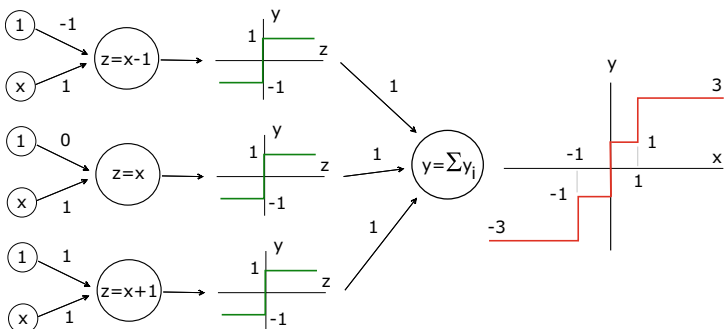


Fig. 1.13 Neural networks are capable of generating functions to fit data regardless of the dimension of the space and the nonlinearity of the problem. In this example, three neurons of the simplest Rosenblatt’s perceptron consisting of a step function are used to generate a local linear function. This function is obtained by simply changing the weights of the bias and adding the results of the three neurons with equal weights. Other more complex functions may be obtained with different weights. Furthermore, the firing step function may be changed by generally better choices as the ReLU or the sigmoid functions

where we note that the first square bracket corresponds to the last layer, the second to the previous one, and so on, until the term in curly brackets addressing the specific network variable. The procedures had issues with exploding or vanishing gradients (especially with sigmoid and hyperbolic tangent activations), but several improvements in algorithms (gradient clipping, regularization, skipping of connections, etc.), have resulted in efficient algorithms for many hidden layers. The complex improvement techniques, with an important number of “tweaks” to make them work in practical problems, is one of the reasons why “canned” libraries are employed and recommended (Fig. 1.13).

1.2.2.8 Bayesian Neural Networks (BNN)

A Bayesian Neural Network (BNN) is a NN that uses probability and the Bayes theorem relating conditional probabilities

$$p(\mathbf{z}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{p(\mathbf{x})} \quad (1.45)$$

where $p(\mathbf{x}|\mathbf{z}) = p(\mathbf{x} \cap \mathbf{z})/p(\mathbf{z})$. A typical example is to consider a probabilistic distribution of the weights (so we take $\mathbf{z} = \mathbf{w}$) for a given model, or a probabilistic distribution of the output (so we take $\mathbf{z} = \mathbf{y}$) not conditioned to a specific model. These choices can be applied in uncertainty quantifications (Olivier et al. 2021), with metal fatigue a typical application case (Fernández et al. 2022a; Bezazi et al. 2007). Given the complexity in passing analytical distributions through the NN, sampling is often performed through Monte Carlo approaches. The purpose is to learn the mean and standard deviations of the distributions of the weights, assuming they follow a normal distribution $w_i \approx \mathcal{N}(\mu_i, \sigma_i^2)$. For the case of predicting an output y , considering one weight, the training objective is to maximize the probability of the training data for the best prediction, or minimize the likelihood of a bad prediction as

$$\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^* = \arg \min_{\boldsymbol{\mu}, \boldsymbol{\Sigma}} \sum_{\forall \mathbf{x}_i, y_i} \mathcal{L}(f(\mathbf{x}_i; \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), y_i)) + \text{KL}[p(\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})), p(\mathcal{N}(0, 1))] \quad (1.46)$$

where $\text{KL}(p_1, p_2)$ is the Kullback–Leibler divergence regularization for the probabilities p_1 and p_2 explained before, \mathcal{L} is the loss function and $f(\mathbf{x}_i; \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}))$ is the function prediction for y from data \mathbf{x}_i , assuming a distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. With the learned optimal parameters $\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*$, the prediction for new data \mathbf{x} is

$$y = \frac{1}{K} \sum_{k=1}^K f(\mathbf{x}; \mathcal{N}_k(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*)) \quad (1.47)$$

where the $\mathcal{N}_k(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*)$ are the numerical evaluations of the normal distributions for the obtained parameters.

1.2.2.9 Convolutional Neural Networks (CNNs)

Although a Convolutional Neural Network (CNN) is a type of FFNN, they were formulated with the purpose of classifying images. CNNs combine one or several convolution layers combined with pooling layers (for feature extraction from images) and with normal final FFNN layers for classification (Fig. 1.14). Pooling is also named subsampling since performing averaging or extracting the maximum of a

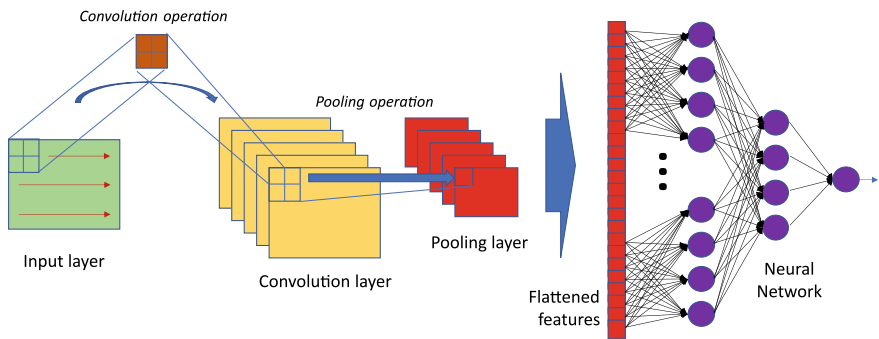


Fig. 1.14 Typical structure of a CNN, including one convolution layer, one pooling layer, a flattened layer of features, and a FFNN

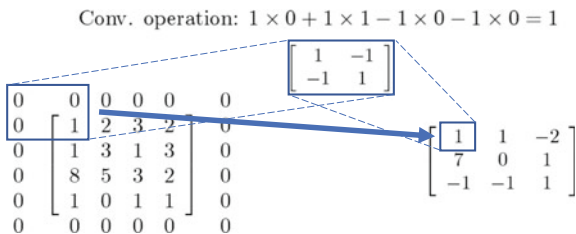


Fig. 1.15 Convolutional network layer with depth 1, stride length 2 (the filter patch moves 2 positions at once) and edge padding 1 (the boundary is filled with one row and column of zeroes). Pooling is similar, but usually selects the maximum or average of a moving pad to avoid correlation of features with location

patch are the typical operations. In the convolutional layers, input data has usually several dimensions, and they are filtered with a moving patch array (also named kernel, with a specific stride length and edge padding; see Fig. 1.15) to reduce the dimension and/or to extract the main characteristics of, or information from, the image (like looking at a lower resolution version or comparing patterns with a reference). Each padding using a patch over the same record is called a channel, and successive or chained paddings are called layers, Fig. 1.15. The same padding, with lower dimension, may be applied over different sample dimensions (a volume). In essence, the idea is similar to the convolution of functions in signal processing to extract information from the signal. Indeed this is also an application of CNN. The structure of CNNs have obvious and interesting applications in multiscale modeling in materials science, and in constitutive modeling (Yang et al. 2020; Frankel et al. 2022), and thus also in determining material properties (Xie and Grossman 2018; Zheng et al. 2020), behavior prediction (Yang et al. 2020), and obviously in extracting microstructure information from images (Jha et al. 2018).

1.2.2.10 Recurrent Neural Networks (RNN)

RNNs are used for sequences of events, so they are extensively used in language processing (e.g., in “Siri” or translators from Google), and they are effective in unveiling and predicting sequences of events (e.g., manufacturing) or when history is important (path-dependent events as in plasticity Mozaffar et al. 2019; Li et al. 2019; du Bos et al. 2020). In Fig. 1.16, a simple RNN is shown with $^t\mathbf{h}$ representing the history variables, such that the equations of the RNN are

$${}^{t+1}\mathbf{h} = \mathbf{f}_h^l(\mathbf{W}_h {}^t\mathbf{h} + \mathbf{W}_x {}^t\mathbf{x} + \mathbf{b}) \tag{1.48}$$

$${}^t\mathbf{o} = \mathbf{f}_o^l(\mathbf{W}_h {}^t\mathbf{h} + \mathbf{W}_x {}^t\mathbf{x} + \mathbf{b}) \tag{1.49}$$

$${}^t\mathbf{y} = \mathbf{f}_y^l(\mathbf{W}_h {}^t\mathbf{h} + \mathbf{b}) \tag{1.50}$$

The unfolding of a RNN allows for a better understanding of the process; see Fig. 1.16. Following our previous seismic example, they can be used to study the prediction of

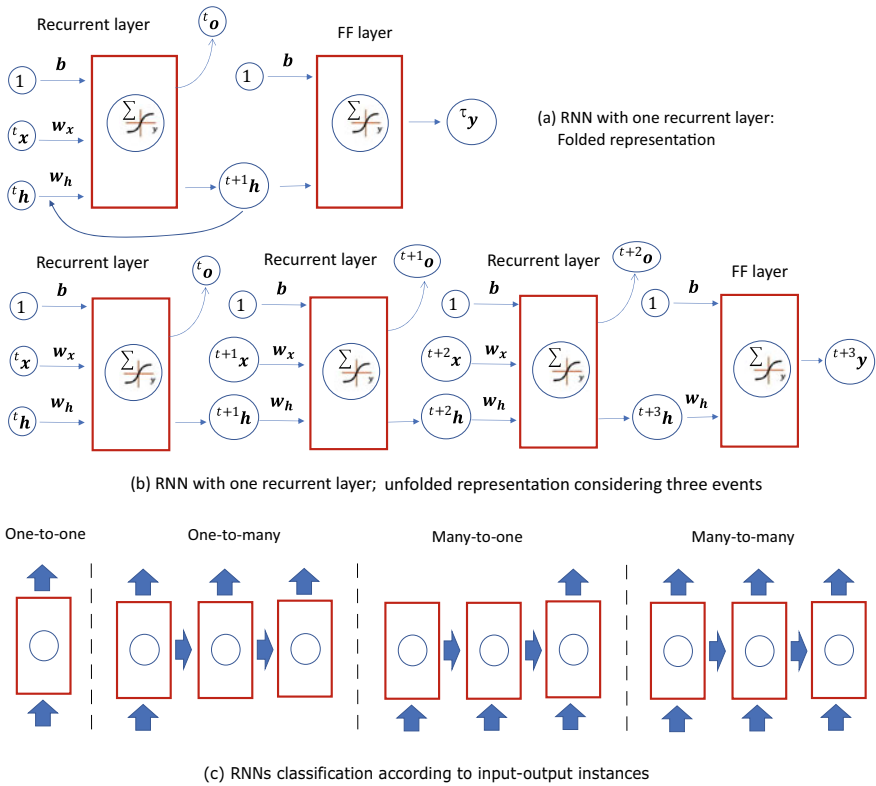


Fig. 1.16 Recurrent Neural Network. **a** Folded representation, **b** unfolded representation considering three events, and **c** classification according to the input–output instances considered

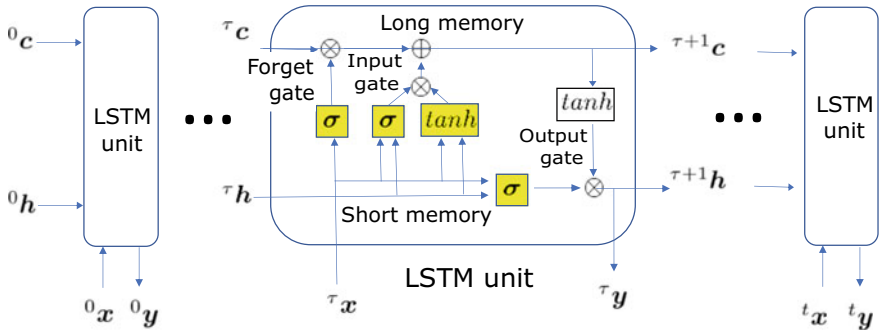


Fig. 1.17 A LSTM RNN, including long and short memory, and forget, input, and output gates. σ is the sigmoid function, colored boxes are typical NN layers, \tanh is the hyperbolic tangent, \otimes and \oplus and \tanh are componentwise operations

new earthquakes from previous history; see, for example, Panakkat and Adeli (2009), Wang et al. (2017). A RNN is similar in nature to a FFNN, and is frequently mixed with FF layers, but recycling some output at a given time or event for the next time(s) or event(s). RNNs may be classified according to the number of related input–output instances as one-to-one, one-to-many (one input instance to many output instances), many-to-one (e.g., classifying a voice or determining the location of an earthquake), and many-to-many (translation into a foreign language); see Fig. 1.16. A frequent ingredient in RNN are “gates” (e.g., in Long Short-Term Memory (LSTM), see Fig. 1.17) to decide which data is introduced, output, or forgotten.

1.2.2.11 Generative Adversarial Networks (GAN)

A Generative Adversarial Network (GAN) (Goodfellow et al. 2020) is a type of ML based on game theory (sum-zero game where one agent’s benefit is the other agent’s loss) with the purpose to learn the probability distribution of the set of training samples (i.e. to solve the generative modeling problem). Although different algorithms have been presented within the GAN paradigm, most are based on NN agents, consisting of a generative NN and a discriminative NN. These NNs have opposite targets. The generative NN tries to fool the discriminative NN, whereas the discriminative NN tries to distinguish original (true) data from generated data presented by the generative NN. With successive events, both NNs learn—the generative NN learns how to fool the other NN, and the discriminative NN how not to be fooled. The type of NN depends on the problem at hand. For example when distinguishing images, a CNN is typically used. In this case, for example, in the falsification of photographs (*deepfake* Yadav and Salmani 2019), several images of a person are presented and the discriminator has to distinguish if they are actual pictures or manufactured photos. This technology is used to generate fake videos, and to detect them (Duarte et al. 2019; Yu et al. 2022) and is used in CAE tasks like the reconstruction of turbulent

velocity fields (by comparing images) (Deng et al. 2019). GANs are also used in the generation of compliant designs, for example in the aeronautical industry (Shu et al. 2020), and also to solve differential equations (Yang et al. 2020; Randle et al. 2020). A recent overview of GANs may be found in Aggarwal et al. (2021).

1.2.2.12 Ensemble Learning

While NNs may bring accurate predictions through extensive training, obtaining such predictions may not be computationally efficient. Ensemble learning consists of employing many low-accuracy but efficient methods to obtain a better prediction through a sort of averaging (or voting). Following our seismic vulnerability example, it would be like asking several experts to give a fast opinion (for example just showing them a photograph) about the vulnerability of a structure or a site, instead of asking one of them to perform a detailed study of the structure (Giacinto et al. 1997; Tang et al. 2022). The methods used may be, for example, shallow NNs and decision trees.

1.3 Constraining to, and Incorporating Physics in, Data-Driven Methods

ML usually gives no insight into the physics of the problem. The classical procedures are considered “black boxes”, with inherent positive (McCoy et al. 2022) and negative (Gabel et al. 2014) attributes. While these black boxes are useful in applications to solve classical fuzzy problems where they have been extensively applied in economy, image or speech recognition, pattern recognition, etc. they have inherently several drawbacks regarding use in mechanical engineering and applied sciences. The first drawback is the large amount of data they require to yield relevant predictions. The second one is the lack of fulfillment of basic physics principles (e.g., the laws of thermodynamics). The third one is the lack of guarantees in the optimality or uniqueness of the prediction, or even guarantees in the reasonableness of the predicted response. The fourth one is the computational cost, if including training, when compared using classical methods. Although once trained, the use may be much faster than many classical methods. Probably, the most important drawback is the lack of physical insight into the problem, because human learning is complex and needs a detailed understanding of the problem to seek creative solutions to unsolved problems. Indeed, in contrast to “unexplainable” AI, now also eXplainable Artificial Intelligence (XAI) is being advocated (Arrieta et al. 2020).

ML may be a good avenue to obtain engineering solutions, but to yield valuable (and reliable), scientific answers, physics principles need to be incorporated in the overall procedure. To this end, the predictions and learning of the previously overviewed methods, or other more elaborated ones, should be restricted to solution subsets that do fulfill all the basic principles. That is, conservation of energy, of linear

momentum, etc. should be fulfilled. When doing so, we use data-driven physics-based machine learning (or modeling) (Ströfer et al. 2018), or “gray-box” modeling (Liu et al. 2021; Asgari et al. 2021; Regazzoni et al. 2020; Rogers et al. 2017). The simplest and probably most used method to impose such principles (an imposition called “whitening” or “bleaching” Yáñez-Márquez 2020) is the use of penalties and Lagrange multipliers in the cost function (Dener et al. 2020; Borkowski et al. 2022; Rao et al. 2021; Soize and Ghanem 2020), but there are many options and procedures to incorporate physics either in the data or in the learning (Karpatne et al. 2017). The resulting methods and disciplines which mix data science and physical equations are often referred to as Physics Based Data Science (PBDS), Physics-Informed Data Science (PIDS), Physics-Informed Machine Learning (PIML) (Karniadakis et al. 2021; Kashinath et al. 2021), Physics Guided Machine Learning (PGML) (Pawar et al. 2021; Rai and Sahu 2020), or Data-Based Physics-Informed Engineering (DBPIE).

In a nutshell, data-based physically informed ML allows for the use of data science methods without most of the shortcomings of physics-uninformed methods. Namely, we do not need much data (Karniadakis et al. 2021), solutions are often meaningful, the results are more interpretable, the methods much more efficient, and the number of meaningless spurious solutions is substantially smaller. The methods are no longer a sophisticated interpolation but can give predictions outside the domain given by the training data. In essence, we incorporate the knowledge acquired in the last centuries.

In PBDS, meaningful internal variables play a key role. In classical engineering modeling, as in constitutive modeling, variables are either external (position, velocity, and temperature) or internal (plastic or viscous deformations, damage, and deformation history). The external variables are observable (common to all methods), whereas the internal variables, being non-observable, are usually based on assumptions to describe some internal state. Here, a usual difference with ML methods is that a physical meaning is typically assigned to internal variables in classical methods, but for example when using NNs, internal variables (e.g., those in hidden layers) have typically no physical interpretation. However, the sought solution of the problem relates external variables both through physical principles or laws and through state equations. To link both physical principles and state equations, an inherent physical meaning is therefore best given (or sought) for the internal ML variables (Carleo et al. 2019; Vassallo et al. 2021). Physical principles are theoretical, of general validity, and unquestioned for the problem at hand (e.g., mass, energy, momentum conservation, and Clausius-Duhem inequality), whereas state equations are a result of assumptions and observations at the considered scales, leading to equations tied to some conditions, assumptions, and simplifications of sometimes questionable generality and of more phenomenological nature.

In essence, the possible ML solutions obtained from state equations must be restricted to those that fulfill the basic physical principles, constituting the physically viable solution manifold, and that is often facilitated by the proper selection of the structure of the ML method and the involved internal variables. These physical constraints may be incorporated in ML procedures in different ways, depending on the analysis and the ML method used, as we briefly discuss below (see also an example in Ayensa Jiménez 2022).

1.3.1 Incorporating Physics in, and Learning Physics From, the Dataset

An objective may be to discover a hidden physical structure in data or physical relations in data (Chinesta et al. 2020). One purpose may be to reduce the dimension of the problem by discovering relations in data that lead to the reduction of complexity (Alizadeh et al. 2020; Aletti et al. 2015). This is similar to calculating dynamical modes of displacements (Bathe and Wilson 1973; Bathe 2006) or to discover the invariants when relating strain components in hyperelasticity (Weiss et al. 1996; Bonet and Wood 1997). Another objective may be to generate surrogate models (Bird et al. 2021; Straus and Skogestad 2017; Jansson et al. 2003; Liu et al. 2021) to discover which variables have little relevance to the physical phenomenon, or quantifying uncertainty in data (Chan and Elsheikh 2018; Trincherio et al. 2018; Abdar et al. 2021; Zhu et al. 2019). Learning physics from data is in essence a data mining approach (Bock et al. 2019; Kamath 2001; Fischer et al. 2006). Of course, this approach is always followed in classical analysis when establishing analytical models, for example when neglecting time effects for quasi-stationary problems, or when reducing the dimension of 3D problems to plane stress or plane strain conditions. However, ML seeks an unbiased automatic approach to the solution of a problem.

1.3.2 Incorporating Physics in the Design of a ML Method

A natural possibility to incorporate physics in the design of the ML method is to impose some equations, in some general form, onto the method, and the purpose is to learn some of the freedom allowed by the equations (Tartakovsky et al. 2018). That is the case when learning material parameters (typical in Materials Science informatics, Agrawal and Choudhary 2016; Vivanco-Benavides et al. 2022; Stoll and Benner 2021), selecting specific functions from possibilities (e.g., selecting hardening models or hyperelastic models from a library of functions, Flaschel et al. 2021, 2022), or learning corrections of models (e.g., deviations of the “model” from reality).

Physics in the design of the ML procedure may also be incorporated by imposing some specific meaning to the hidden variables (introducing physically meaningful internal variables as the backstress in plasticity) or the structure (as the specific existence and dependencies of variables in the yield function) (Ling et al. 2016). Doing so, the resulting learned relations may be better interpreted and will be in compliance with our previous knowledge (Abueidda et al. 2021; Miyazawa et al. 2019; Zhan and Li 2021).

A large amount of ML methods in CAE are devoted to learning constitutive (or state) equations (Leygue et al. 2018), with known conservation principles and kinematic relations (equilibrium and compatibility), as well as the boundary conditions (González et al. 2019b; He et al. 2021). In essence, we can think of a “physical man-

ifold” and a “constitutive manifold”, and we seek the intersection of both for some given actions or boundary and initial conditions (Ibañez et al. 2018; He et al. 2021; Ibañez et al. 2017; Nguyen and Keip 2018; Leygue et al. 2018). Autoencoders are a good tool to reduce complexity and filter noise (He et al. 2021). Other methods are devoted to inferring the boundary conditions or material constitutive inhomogeneities (e.g., local damage) assuming that the general form of the constitutive relations is known (this is a ML approach to the classical inverse problem of damage/defect detection).

Regarding the determination of the constitutive equations, the procedure may be purely data-driven (without the explicit representation of a constitutive manifold or constitutive relations, i.e. “model-free” Kirchdoerfer and Ortiz 2016, 2017; Eggersmann et al. 2021a; Conti et al. 2020; Eggersmann et al. 2021b) or manifold-based, in which case a constitutive manifold is established as a data-based constitutive equation. In the model-free family, we assume that a large amount of data is known, so a material data “point” is always close to the physical manifold (see Fig. 1.18 left). Then, while these techniques may be considered within the ML family, they are more data-driven deterministic techniques (raw data is employed directly, no constitutive equation is “learned”). In the manifold-based family (Fig. 1.18, center and right), the manifold may be explicit (e.g., spline-based, Sussman and Bathe 2009; Crespo and Montáns 2019; Latorre and Montáns 2017; Crespo et al. 2017; Coelho et al. 2017) or implicit (discrete or local, e.g., Lopez et al. 2018; Ibañez et al. 2020; Meng et al. 2018; Ibañez et al. 2017). This is a family of methods for which the objective is to learn the state equations from known (experimental or analytical) data points, probably subject to some physics requirements (as integrability). Within this approach, once the manifold is established, the computation of the prediction follows a scheme very similar to the use of classical methods (Crespo et al. 2017).

Remarkably, in some Manifold Learning approaches, physical requirements (which may include, or not, physical internal variables, Amores et al. 2020) may result in a substantial reduction of the experimental data needed (Latorre et al. 2017; Amores et al. 2021) and of the overall computational effort, resulting also in an

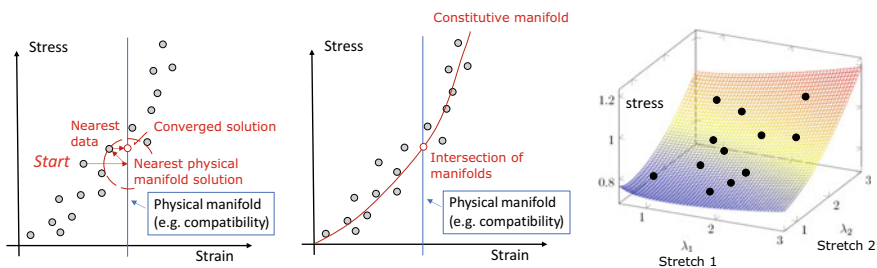
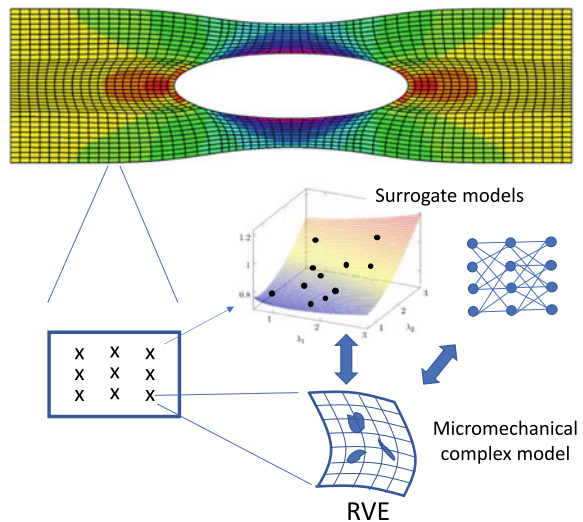


Fig. 1.18 Data-based constitutive modeling. Left: purely data-driven technique, where no constitutive manifold is directly employed. Instead, the closest known data point is located and is used to compute the solution. Center, right: constitutive (e.g., stress–strain) data points are used to compute a constitutive manifold (which may include uncertainty quantification), which is then employed to compute the solution in a classical manner

increased interpretability of the solution. An important class of problems where ML in general, and Manifold Learning approaches in particular, are often applied with important success, is the generation of surrogate models for multiscale problems (Peng et al. 2021; Yan et al. 2020; White et al. 2019; El Said and Hallett 2018; Alber et al. 2019; Brunton and Kutz 2019). The solutions of nonlinear multiscale problems, in particular those which use Finite Element based computational homogenization (FE squared) (FE2) techniques, are still very expensive, because at each integration point, a FE problem representing the Representative Volume Element (RVE) must be considered (Fish et al. 2021; Arbabi et al. 2020; Fuhs et al. 2021). Then, surrogate models which represent the equivalent behavior at the continuum level are extremely useful (Fig. 1.19). These surrogate models may be obtained using different techniques. The use of Neural Networks is one option (Wang and Sun 2018; Wang et al. 2020). Then the dataset for the training is obtained from repeated off-line simulations with different loading and boundary conditions at different deformation levels and with different loading histories (Logarzo et al. 2021). Another option is to use surrogate models based on the equivalence of physical quantities as stored and dissipated energies (Crespo et al. 2020; Miñano and Montáns 2018). Reduced Order Methods are also important, especially in nonlinear path-dependent procedures to determine the main internal variables or simplest representation giving sufficient accuracy (Singh et al. 2017; Rocha et al. 2020). An important aspect in surrogate modeling is the possibility of inversion of the map (Haghighat et al. 2021; Raissi et al. 2019; Haghighat and Juanes 2021), which is crucial when prediction is not the main purpose of the machine learning procedure but the main objective is to learn about the material or its spatial distribution. The use of autoencoders can be effective if decompression is fundamental in the process (Kim et al. 2021; Bastek et al. 2022; Xu et al. 2022; Jung et al. 2020).

Fig. 1.19 Surrogate modeling representing the micromechanics. Examples are numerical manifolds or Neural Networks



1.3.3 Data Assimilation and Correction Methods

The use of ML models, as when using any model (including classical analytical models; see, for example, Bathe 2006), may result in a significant error in the prediction of the actual physical response. This error may be produced either by insufficient data (or insufficient quality of the data because of noise or lack of completeness), or by inaccuracy of the model (e.g., due to too few layers in a NN or erroneous or oversimplifying assumptions) (Haik et al. 2021). Then problems are how to incorporate new data (labeled or unlabeled) into the model (Buizza et al. 2022), how to enrich the model to improve the predictions (Singh et al. 2017), and how to augment physical models with machine-learned bias (Volpiani et al. 2021) (hybrid models). These problems are typically encountered in dynamics (Muthali et al. 2021), and the solutions are often similar to those employed in control theory (Rubio et al. 2021), as the use of Kalman methods (Zhang et al. 2020). Machine learning techniques may be used for self-learning complex physical phenomena as the sloshing of fluids (Moya et al. 2020). In essence, the proposal here is to assume that there is a model-predicted response $\mathbf{y}^{\text{model}}$ and a true (say “experimental”) response \mathbf{y}^{exp} (Moya et al. 2022). The difference is the error to be corrected, namely $\mathbf{y}^{\text{corr}} = \mathbf{y}^{\text{exp}} - \mathbf{y}^{\text{model}}$. This error is corrected in further predictions by assuming that there is an indeterminacy either in the input data (statistical error) or in the model (some unknown variables that are not being considered). Note that the statistical error case is conceptually similar to the quantification of uncertainty. In case the model needs corrections, some formalism may be employed to introduce physics corrections to learned models. For example, correcting dissipative behavior in assumed (hyper)elastic behavior (or vice versa). In case there are some indeterminacies in the model, we can assume that the model is of the form

$$\mathbf{y} = f(\mathbf{x}; \mathbf{w}, \boldsymbol{\omega}) \quad (1.51)$$

where the \mathbf{w} are the parameters determined previously (e.g., during the usual model learning process and now fixed) and $\boldsymbol{\omega}$ are the parameters correcting the model by minimizing the error. This model correction process using new data is called data assimilation. In Dynamic Data-Driven Application Systems (DDDAS), the concepts of Digital Twins and Hybrid Twins are employed. A Digital Twin (Glaessgen and Stargel 2012) is basically a virtual (sometimes comprehensive) model which is used as a replication of a system in real life. For example, a Formula-1 simulator, Mayani et al. 2018, or a spacecraft simulator, Ye et al. 2020; Wang 2020 may be considered a Digital Twin (Luo et al. 2020). A Digital Twin serves as a platform to try new solutions when it is difficult or expensive to try them in the actual physical system. Digital Twins are increasingly used in industry in many fields (Bhatti et al. 2021; Garg and Panigrahi 2021; Burov and Burova 2020). This virtual platform may contain classical analytical models, data-driven models, or a combination of both (which is currently very usual in complex systems). The concept of Hybrid Twin (Chinesta et al. 2020) (or self-learning digital twin, Moya et al. 2020) is a step forward, which mixes the virtual/digital twin model with model order reductions and parametrized solutions. The purpose is to have a twin in real time, which may be used to predict

the behavior of a system in advance and correct the system (Moya et al. 2022) or take any other measure; that is, in essence to control a complex physical system. The dynamic equation of the Hybrid Twin is

$$\dot{\mathbf{X}}(t; \boldsymbol{\mu}) = \mathbf{A}(\mathbf{X}, t; \boldsymbol{\mu}) + \mathbf{B}(\mathbf{X}, t) + \mathbf{C}(t) + \mathbf{R}(t) \quad (1.52)$$

where the $\boldsymbol{\mu}$ are the model parameters, $\mathbf{A}(\mathbf{X}, t; \boldsymbol{\mu})$ is the (possibly analytical) model contribution given those parameters (a linear model would be $\mathbf{A}(\boldsymbol{\mu})\mathbf{X}$) (Sancarlos et al. 2021), $\mathbf{B}(\mathbf{X}, t)$ is a data-based correction to the model (a continuous update from measurements), $\mathbf{C}(t)$ are the external actions, and $\mathbf{R}(t)$ is the (unbiased and unpredictable) noise. We use the word “hybrid” (Champaney et al. 2022) because analytical and data-based approaches are employed. Hybrid Twins have been applied in various fields, for example in simulating acoustic resonators (Martín et al. 2020).

1.3.4 ML Methods Designed to Learn Physics

A different objective from incorporating physics in the ML method is to use a ML method to learn physics. One example would be to learn constitutive equations without prior (or with minimal) assumptions—a case that is similar to those discussed above but, for example, without neglecting a priori the influence of some terms or without assuming the nature of the constitutive equation (for example, not assuming elasticity, plasticity, or other). Another example is to learn new physical or fundamental evolution equations in nature. A successful (and quite simple) case is the Sparse Identification of Physical Systems, in particular the Sparse Identification of Nonlinear Dynamics (SINDy) (Brunton et al. 2016; Rudy et al. 2017). In this approach, the nonlinear problem

$$\dot{\mathbf{X}} = \mathbf{A}(\mathbf{X}) \quad (1.53)$$

is re-written as

$$\dot{\mathbf{X}} = \boldsymbol{\Theta}(\mathbf{X}) \boldsymbol{\Xi} \quad (1.54)$$

where $\boldsymbol{\Xi}$ is a sparse matrix of dynamical coefficients and $\boldsymbol{\Theta}(\mathbf{X})$ contains a library of functions evaluated at \mathbf{X} . In the Lorenz System shown in Fig. 1.20 (Brunton et al. 2016), $\boldsymbol{\Theta}(\mathbf{X})$ involves a set of nonlinear polynomial combinations of the components of \mathbf{X} . The purpose here is to obtain the possibly simplest yet accurate description (the parsimonious model) in terms of the expansion functions, and this is performed by the technique of sparse regression, which promotes sparsity in underdetermined least squares regression by replacing the norm-2 Tikhonov regularization by a norm-1 penalization (Tibshirani 1996), although in Brunton et al. (2016) the authors used a slightly different technique. The optimal penalty may be obtained by minimizing a cross-validation error (i.e. the solution which is accurate but avoids overfitting). The method has been applied to a variety of physics problems to determine their differential equations (Rudy et al. 2017). Similar approaches are Physics-Informed Spline Learning (PiSL) (Sun et al. 2021), which represents an improvement for data

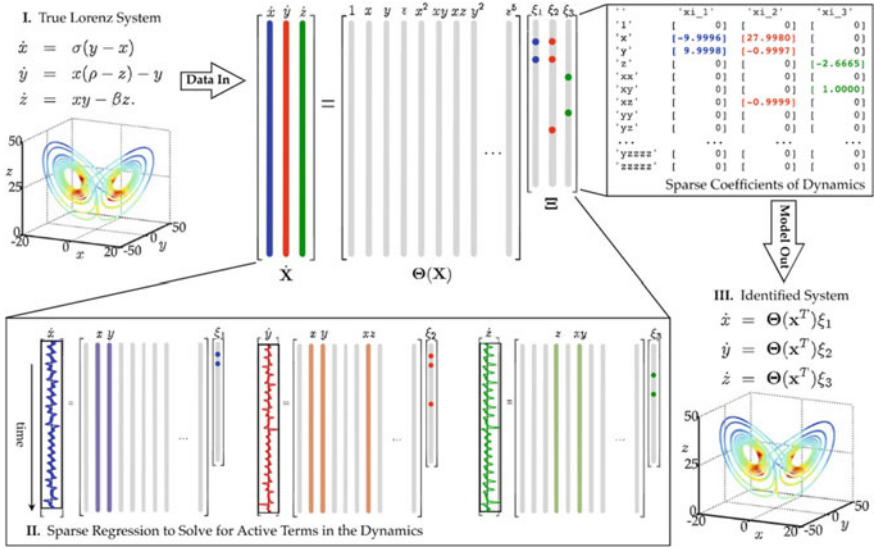


Fig. 1.20 Sparse Identification of Nonlinear Dynamics. Case of Lorenz System. Reproduced from Brunton et al. (2016)

representation allowing for explicit derivatives and uses alternating direction optimization with adaptive Sequential Threshold Ridge regression (STRidge) (Rudy et al. 2017) for promoting sparsity, and also more classical genetic and symbolic regression procedures (Searson 2009; Schmidt and Lipson 2009, 2010). An overview of these techniques and others may be found in Brunton and Kutz (2022); see also Zhang and Liu (2021) for a progressive approach for considering uncertainties.

These approaches, as the SINDy type, can trivially address the correction given by an imperfect modeling (i.e. the Hybrid Twin). It simply suffices to consider a correction in Eq. (1.53)

$$\dot{\mathbf{X}} - \mathbf{A}(\mathbf{X}) = \mathbf{B}(\mathbf{X}) \tag{1.55}$$

where $\mathbf{B}(\mathbf{X})$ is the measured discrepancy to be corrected between the results obtained from the inexact model and the experimental results. As performed in mathematics and physics, the key for simplification and possible linearization of a complex problem consists of finding a proper (possibly reduced) space of (possibly transformed) input variables to re-write the problem. As mentioned, NNs, in particular autoencoders, can be used to find the space, to which, thereafter, a SINDy approach may be applied to create a Digital or Hybrid Twin (Champion et al. 2019). These mixed NN approaches have also been employed in multiscale physics transferring learning through scales by increasingly deep and wide NNs (Liu et al. 2020), also employing CNNs (Liu et al. 2022). Of course, Dynamic Mode Decomposition (DMD) (Schmid 2010; Tu 2013; Schmid 2011; Jovanović et al. 2014; Demo et al. 2018), a procedure to determine coupled spatio-temporal modes for nonlinear problems based on Koopman (composite operator) theory (Williams et al. 2015), is also used for incorporating data

into physical systems, or determining the physical system equations themselves. The idea is to obtain two sets (“snapshots”) of spatial measurements separated by a given Δt , namely ${}^t\mathbf{X}$ and ${}^{t+\Delta t}\mathbf{X}$. Then, the eigenvectors of $\mathbf{A} = {}^{t+\Delta t}\mathbf{X} {}^t\mathbf{X}^{-1}$, where ${}^t\mathbf{X}^{-1}$ is the pseudoinverse, are the best regressors to the linear model, that is, the minimum-squares best fit of the nonlinear model, compatible with the snapshots. In practice, the \mathbf{A} matrix is usually not computed because working with the SVD of \mathbf{X} is more efficient (Proctor et al. 2016).

Other techniques to discover physical relations (or nonlinear differential equations), as well as simultaneously obtain physical parameters and fields, are physics-informed NN (PINN) (Raissi and Karniadakis 2018; Raissi et al. 2019; Pang et al. 2019; Yang et al. 2021). For example, using neural networks, the viscosity, the density, and the pressure, with the velocity field in time may be obtained assuming the Navier–Stokes equations as background and employing a NN as the learning engine to match snapshots. Moreover, these methods may be combined with time integrators for obtaining the nonlinear parameters of any differential equation, including higher derivatives, just from discretized experimental snapshots (Meng et al. 2020; Zhang et al. 2020). Other applications include inverse problems in discretized conservative settings (Jagtap et al. 2020).

1.3.4.1 Deep Operator Networks

While it is very well known that the so-called universal approximation theorem guarantees that a neural network can approximate any continuous function, it is also possible to approximate continuous operators by means of neural networks (Chen and Chen 1995). Based on this fact, Lu and coworkers have proposed the Deep Operator Networks (DeepONets) (Lu et al. 2021).

A DeepONet typically consists of two different networks working together: one to encode the input function at a number of measurement locations (the so-called branch net) and a second one (the trunk net) to encode the locations for the output functions. Assume that we look forward to characterize an operator $F : X \rightarrow Y$, with X, Y two topological spaces. For any function $x \in X$, this operator produces $G = F(x)$, the output function. For any point y in the domain of $F(x)$, $G(y)$ is a real number. A DeepONet thus learns from pairs (x, y) to produce the operator. However, for an efficient training, the input function x is sampled at discrete spatial locations.

In some examples, DeepONets showed very small generalization errors and even exponential error convergence with respect to the training dataset size. This is however not yet fully understood. DeepONets have been applied, for example, to predict crack paths in brittle materials (Goswami et al. 2022), instabilities in boundary layers (Di Leoni et al. 2021), and the response of dynamical systems subjected to stochastic loadings (Garg et al. 2022).

Recently, DeepONets have been generalized by parameterizing the integral kernel in Fourier space, giving rise to the so-called Fourier Neural Operators (Li et al. 2020). These networks have also gained a high popularity, and have been applied to weather forecasting, for instance (Pathak et al. 2022).

1.3.4.2 Neural Networks Preserving the Physical Structure of the Problem

Within the realm of PIML approaches, a new family of methods has recently been proposed. The distinctive characteristic is that these new techniques see the supervised learning process as a dynamical system as

$$\dot{\mathbf{z}} = \mathbf{f}(\mathbf{z}, t), \text{ with } \mathbf{z}(0) = \mathbf{z}_0 \quad (1.56)$$

with \mathbf{z} being the set of variables governing the problem. The supervised learning problem will thus be to establish \mathbf{f} in such a way as to reach an accurate description of the evolution of the variables. By formulating the problem in this way, the analyst can use the knowledge already available, and established over centuries, on dynamical systems. For instance, adopting a Hamiltonian perspective on the dynamics and enforcing \mathbf{f} to be of the form

$$\dot{\mathbf{z}} = \mathbf{L}\nabla H \quad (1.57)$$

where \mathbf{L} is the classical (skew-symmetric) symplectic matrix, which ensures that the learnt dynamics will conserve energy, because it is derived from the Hamiltonian H . Many recent references have exploited this approach, either in Hamiltonian or Lagrangian frameworks (Greydanus et al. 2019; Mattheakis et al. 2022; Cranmer et al. 2020). If the system of interest is dissipative—which is, by far, most frequently the case—a second potential must be added to the formulation as

$$\dot{\mathbf{z}} = \mathbf{L}\nabla H + \mathbf{M}\nabla S \quad (1.58)$$

where S represents the so-called Mathieu potential. To ensure the fulfillment of the first and second principles of thermodynamics, an additional restriction (the so-called degeneracy conditions) must be imposed, i.e.

$$\mathbf{L}\nabla S = \mathbf{M}\nabla H = \mathbf{0} \quad (1.59)$$

These equations essentially state that entropy has nothing to do with energy conservation and, in turn, energy potentials have nothing to do with dissipation. The resulting NN formulations produce predictions that comply with the laws of thermodynamics (Hernández et al. 2021, 2022).

1.4 Applications of Machine Learning in Computer Aided Engineering

In this section we describe some applications of machine learning in CAE. The main purpose is to briefly focus on a variety of topics and ML approaches employed in several fields, but not to give a comprehensive review. Hence, given the vast literature already available, developed in the last few years, many important works have likely

been omitted. However, even though the field of applications is very broad, the main ideas fundamental to the techniques are given in the previous sections.

1.4.1 Constitutive Modeling and Multiscale Applications

The main field of application of machine learning techniques in CAE is ML constitutive modeling, both at the continuum scale and for easing multiscale computations. As previously mentioned, applicable procedures are model-free approaches, data-driven manifold learning, data-driven model selection and calibration, and surrogate modeling. Another interesting application of ML, in particular NNs, is to improve results from coarse FE models without resorting to expensive fine computations, e.g., “zooming” (Yamaguchi and Okuda 2021). There are several reviews of applications of ML in constitutive modeling (specially using NNs), in continuum mechanics (Bock et al. 2019), for soils (Zhang et al. 2021), composites (Zhang and Friedrich 2003; Liu et al. 2021; El Kadi 2006), and material science (Hkdh 1999). An earlier review of NN applications in computational mechanics in general can also be found in Yagawa and Okuda (1996). Below we briefly review some applications.

1.4.1.1 Linear and Nonlinear Elasticity

One of the simplest modeling problems and, hence, one of the most explored ones is the case of elasticity. The linear elastic problem, addressed from a model-free data-driven method is analyzed in Kirchdoerfer and Ortiz (2016), Conti et al. (2018), and even earlier in Wang et al. (2011) for cloths in the animation and design industries. Data-driven nonlinear elasticity is also analyzed in several works (Conti et al. 2020; Stainier et al. 2019; Nguyen and Keip 2018), and applied to soft tissues (González et al. 2020) and foams (Frankel et al. 2022).

In particular, data-driven specific solvers are needed if model-free methods are employed, and some effort is directed to developing such solvers and data structuring methods for the task (Eggersmann et al. 2021a,b; Platzer et al. 2021). Kernel regression is also employed (Kanno 2018).

Another common methodology is the use of data-driven constitutive manifolds (Ibañez et al. 2017), where identification and reduction of the constitutive manifolds allow for a much more efficient approach. NNs are as well used in finite deformation elasticity (Nguyen-Thanh et al. 2020; Wang et al. 2022).

Remarkably, nonlinear elasticity is one of the cases where physics-informed methods are important, because true elasticity means integrable path-independent constitutive behavior, i.e. hyperelasticity. Classical ML methods are not integrable (hence not truly elastic). To fulfill such requirement, specific methods are needed (González et al. 2019b; Chinesta et al. 2020; Hernandez et al. 2021). One of the possibilities is to posit the state variables and a reduced expression of the hyperelastic stored energy (which may be termed as “interpretable” ML models Flaschel et al. 2021). Then, this energy may be modeled, for example, by splines or B-splines. This approach,

based on the Valanis–Landel assumption, was pioneered by Sussman and Bathe for isotropic polymers (Sussman and Bathe 2009) and extended later for anisotropic materials (Latorre and Montáns 2013) like soft biological tissues (fascia, Latorre et al. 2017, skin Romero et al. 2017, heart Latorre and Montáns 2017, muscle Latorre et al. 2018, Moreno et al. 2020), compressible materials (Crespo et al. 2017), auxetic foams (Crespo and Montans 2018; Crespo et al. 2020), and composites (Amores et al. 2021). Polynomials in terms of invariants are also employed, with the coefficients determined by sparse regression (Flaschel et al. 2021). Another approach is to select models from a database, and possibly correct them (González et al. 2019a; Erchiqui and Kandil 2006), or select specific function models for the hyperelastic stored energy using machine learning methods (e.g., NNs) (Flaschel et al. 2021; Vlassis et al. 2020; Nguyen-Thanh et al. 2020). In particular, polyconvexity (to guarantee stability and global minimizers for the elastic boundary-value problem) may also be imposed in NN models (Klein et al. 2022). Anisotropy in hyperelasticity may be learned from data with NNs (Fuhg et al. 2022a).

In material datasets, noise and outliers may be a relevant issue, both regarding accuracy and their promotion of overfitting. Clustering has been employed in model-free methods to assign a different relevance depending on the distance to the solution and using an estimation based on maximum entropy (Kirchdoerfer and Ortiz 2017). For spline-based constitutive modeling, experimental data reduction using stability-based penalizations allows for the use of noisy datasets and outliers avoiding overfitting (Latorre and Montáns 2020).

1.4.1.2 Plasticity, Viscoelasticity, and Damage

ML modeling of nonconservative effects is still in quite an incipient state because path-dependency requires the modeling of latent internal variables and the knowledge of the previous deformation path (González et al. 2021). However, some early works using NNs are available (Panagiotopoulos and Waszczyszyn 1999). The amount of needed data is much larger because the possible deformation paths are infinite, but there are already a relevant number of works dealing with inelasticity. In the case of damage, spline-based What-You-Prescribe is What-You-Get (WYPiWYG) large-strain modeling is available both for isotropic (Miñano and Montáns 2015) and anisotropic materials (Miñano and Montáns 2018). Crack growth in the aircraft industry has also been determined with RNNs (Nascimento and Viana 2020). Of course, ML has been for a long time applied to model fatigue (Lee et al. 2005).

Plasticity is probably the most studied case of the nonconservative behaviors (Waszczyszyn and Ziemiański 2001). For the case of data-driven (model-free) “extended experimental constitutive manifolds” including internal variables, the LARge Time INcrement (LATIN) method (solving by separating the constitutive and compatibility/equilibrium sets and looking for the intersection) has been successfully used (Ladevèze et al. 2019); see also Ibañez et al. (2018).

Data-driven model-free techniques in plasticity and viscoelasticity have been developed using more general history variables (like the history of stresses or strains

as typically pursued for hereditary models) (Eggersmann et al. 2019; Ciftci and Hackl 2022). FFNNs with PODs have been employed to fit several plasticity stress–strain behaviors. NNs are also used to replace the stress integration approaches in FE analysis of elastoplastic models (Jang et al. 2021). In general, RNNs (Mozaffar et al. 2019; Borkowski et al. 2022) and CNNs (Abueidda et al. 2021) are a good resort for predicting plastic paths, and sophisticated LSTM and Gated Recurrent Unit (GRU) schemes have been reported to give excellent predictions even for complex paths (Wang et al. 2020).

In materials science, ML is employed to predict the cyclic stress–strain behavior depending on the microstructure of the material obtained from electron backscatter diffraction (EBSD) analysis. The shape of the yield function can also be determined by employing sparse regression from a strain map and the cell load in a non-homogeneous test (like considering a plate with holes) (Flaschel et al. 2022). A mixture of analytical formulas and FFNN machine learning has been employed to replace the temperature- and rate-dependent term of the Johnson–Cook model (Li et al. 2019). In plasticity, physics-based modeling is incorporated by assuming the existence of a stored energy, a plasticity yield function, and a plastic flow rule. These may be obtained by NNs learned from numerical experiments on polycrystal databases, resulting in a more robust ML approach than using the classical black-box ML scheme (Vlassis and Sun 2021). Support Vector Regression (SVR), Gaussian Process Regression (GPR), and NNs have been used to determine data-driven yield functions with the convexity constraints required by the theory (Fuhg et al. 2022b). Automatic hyperparameter (self-)learning has been addressed for NN modeling of elastoplasticity in Fuchs et al. (2021).

1.4.1.3 Fracture

Fracture phenomena may also be modeled using NNs (Theocaris and Panagiotopoulos 1993; Seibi and Al-Alawi 1997) and data-driven model-free techniques (Carrara et al. 2020). Data-driven model extraction from experimental data and knowledge transfer (Goswami et al. 2020) have been applied to obtain predictions in 3D models from 2D cases (Liu et al. 2021). Data-driven approaches are used to enhance fracture paths in simulations of random composites and in model reduction to avoid high fidelity phase-field computations (Guillemot and Dolbow 2020). SVMs and variants have been used for predicting fracture properties, e.g., Yuvaraj et al. (2013), Kulkarni et al. (2011), and so have been other methods like BNN, Genetic Algorithm (GA), and hybrid systems; see, for example, Nasiri et al. (2017), Hoshyar et al. (2020).

1.4.1.4 Multiscale and Composites Modeling

The modeling of complex materials is one of the fields where machine learning may bring about significant advances in CAE (Peng et al. 2021), in particular when

nonlinear behavior is modeled (Jackson et al. 2019). This is particularly the case when the macroscopic behavior or the physical properties depend in a complex manner on a specific microstructure (Fish et al. 2021) or on physics equations and phenomena only seen at a micro- or smaller scale, as atomistic (Caccin et al. 2015; Kontolati et al. 2021; Wood et al. 2019), molecular (Xiao et al. 2020), or cellular (Verkhivker et al. 2020).

ML allows for the simpler implementation of first-principles in multiscale simulations (Hong et al. 2021), describing physical macroscopic properties, like also in chaotic dynamical systems for which the highly nonlinear behavior depends on complex interactions at smaller scales (e.g., weather and climate predictions) (Chattopadhyay et al. 2020). Generating surrogate models to reproduce the observed macroscopic effects due to complex phenomena at the microscale (Wirtz et al. 2015) is often only possible through ML and Model Order Reduction (MOR) (Wang et al. 2020; Yvonnet and He 2007). Even in the simplest cases, ML may substantially speed up the expensive computational costs of classical nonlinear FE2 homogenization techniques (Feng et al. 2022; Wu et al. 2020), allowing for real-time simulations (Rocha et al. 2021). The nonlinear multiscale case is complex because an infinite number of simulations would be needed for a complete general database. However, a reduced dataset may be used to develop a numerical constitutive manifold with sufficient accuracy, e.g., using Numerically EXplicit Potentials (NEXP) (Yvonnet et al. 2013). Material designs are often obtained from inverse analyses facilitated by parametric ML surrogate models (Jackson et al. 2019; Haghghat et al. 2021). In particular, ML may be employed to determine the phase distributions in heterogeneous materials (Valdés-Alonzo et al. 2022).

The modeling of classical fiber-based and complex composite heterogeneous materials often requires multiscale approaches (Pathan et al. 2019; Hadden et al. 2015; Kanouté et al. 2009) because modeling of interactions at the continuum level requires inaccurate assumptions. CNNs are ideal for dealing with the relation of an unstructured RVE with continuum equivalent properties. In particular, ML may be used for dealing with stochastic distributions of constituents (Liu et al. 2022). Modeling of complex properties such as composite phase changes for thermal management in Li-ion batteries may be performed with CNNs (Kolodziejczyk et al. 2021). Indeed, CNNs can also be used for performing an inverse analysis (Sorini et al. 2021). In general, many complex properties and effects observed macroscopically, but through effects mainly attributed to the microscale, are often addressed with different ML techniques, including CNNs, e.g., Field et al. (2021), Nayak et al. (2022), and Koumoulos et al. (2019).

1.4.1.5 Metamaterials Modeling

Metamaterials are architected materials with inner custom-made structure. With the current development of 3D printing, metamaterial modeling and design is becoming an important field (Kadic et al. 2019; Bertoldi et al. 2017; Zadpoor 2016; Barchiesi et al. 2019) because a material with unique salient properties may be designed ad

libitum allowing for a wide range of applications (Surjadi et al. 2019). Their design has evolved from the classical optimization-based approach (Sigmund 2009). ML methods for the design of metamaterials are often used with two objectives. The first objective is to generate simple surrogate models to accelerate simulations avoiding FE modeling to the very fine scale describing the structure, especially when nonlinearities are important. The second objective is to perform analyses using a metamaterial topology parametrization which allows for an effective metamaterial design from macroscopic desired properties. Examples of ML approaches for metamaterials pursuing these two objectives can be found in, e.g., Wu et al. (2020), Fernández et al. (2022b), Zheng et al. (2020), and Wilt et al. (2020).

1.4.2 Fluid Mechanics Applications

Fluid phenomena and related modeling approaches are very rich, spanning from the breakup of liquid droplets under different conditions (Krzeczkowski 1980; Roisman et al. 2018; Liu et al. 2018) to smoke from fires in tunnels (Gannouni and Maad 2016; Wu et al. 2021), emissions from engines (Khurana et al. 2021; Baklacioglu et al. 2019), flow and wake effects in wind turbines (Clifton et al. 2013; Ti et al. 2020), and free surface flow dynamics (Becker and Teschner 2007; Scardovelli and Zaleski 1999). The difficulty in obtaining accurate and efficient solutions, especially when effects at multiple scales are important, has fostered the introduction of ML techniques. We briefly review some representative works.

1.4.2.1 Turbulence Flow Modeling

The modeling of turbulence is an important aspect in the solution of the Navier–Stokes equations of fluid flows. Here ML techniques can be of value.

The ML procedures in turbulence often build on the Reynolds Averaging decomposition, $\mathbf{u}(\mathbf{x}, t) = \bar{\mathbf{u}}(\mathbf{x}) + \tilde{\mathbf{u}}(\mathbf{x}, t)$ which splits the flow $\mathbf{u}(\mathbf{x}, t)$ into an average $\bar{\mathbf{u}}(\mathbf{x})$ time-independent component and a fluctuating component $\tilde{\mathbf{u}}(\mathbf{x}, t)$ with zero average, obtaining the incompressibility conditions $\nabla \cdot \bar{\mathbf{u}} = 0$ and $\nabla \cdot \tilde{\mathbf{u}} = 0$. Then, the Navier–Stokes equations are written in terms of the Reynolds stresses $\rho \overline{\tilde{\mathbf{u}} \otimes \tilde{\mathbf{u}}}$

$$\nabla \bar{\mathbf{u}} \cdot \bar{\mathbf{u}} = \frac{1}{\rho} \nabla(-p\mathbf{I} + 2\mu \nabla^s \bar{\mathbf{u}} - \rho \overline{\tilde{\mathbf{u}} \otimes \tilde{\mathbf{u}}}) \quad (1.60)$$

for which a turbulence closure model is assumed, e.g., eddy viscosity model or the more involved $k - \varepsilon$ (Gerolymos and Vallet 1996) or Spalart–Allmaras models (Spalart and Allmaras 1992). In Eq. (1.60), $\nabla^s \bar{\mathbf{u}}$ is the average deviatoric strain-rate tensor. The framework in Eq. (1.60) gives the two commonly used models: the Reynolds-Averaged Navier–Stokes (RANS) model, best for steady flows (Speziale 1998; Kalitzin et al. 2005), and the Large Eddy Simulations (LES) model, using a subgrid-scale model, thus much more expensive computationally, but best used to predict flow separation and fine turbulence details. RANS closure models have been

explored using ML. For example, the work reported (Zhao et al. 2020) trains a turbulence model for wake mixing using a CFD-driven Gene Expression Programming (an evolutionary algorithm). Physics-informed ML may also be used for augmenting turbulence models, in particular to overcome the difficulties of ill-conditioning of the RANS equations with typical Reynolds stress closures, focusing on improving mean flow predictions (Wu et al. 2018). Results of using ML to improve accuracy of closure models are, for example, given in Wackers et al. (2020), Wang et al. (2017). One of the important problems in modeling turbulence and accelerating full field simulations is to upscale the finer details, e.g., vorticity from the small to the larger scale, using a lower resolution (grid) analysis. These upscaling procedures may be performed by inserting NN corrections which learn the scale evolution relations, greatly accelerating the computations by allowing lower resolution (Kochkov et al. 2021).

1.4.2.2 Shock Dynamics

More accurate and faster shock-capturing by NN has been pursued in Stevens and Colonius (2020), where ML has been applied to improve finite volume methods to address discontinuous solutions of PDEs. In particular, Weighted Essentially Non-Oscillatory Neural Network (WENO-NN) approaches establish the smoothness of the solution to avoid spurious oscillations, still capturing accurately the shock, where the ML procedure facilitates the computation of the optimal nonlinear coefficients of each cell average.

1.4.2.3 Reduced Models for Accelerating Simulations

An important application of ML in fluid dynamics and aerodynamics is the development of reduced order models. In essence, these models capture the main dominant coarse flow structures, with fine structures included and provide a faster, simpler model for analysis, i.e. a surrogate model of similar idea as those used in multiscale analysis. As mentioned previously, there are many techniques used for this task, such as DMD (Schmid et al. 2011; Hemati et al. 2014) or more general POD (Berkooz et al. 1993; Aubry 1991; Rowley 2005), PGD (Dumon et al. 2011; Chinesta et al. 2011), PCA (Audouze et al. 2009), and SVD (Lorente et al. 2008; Braconnier et al. 2011). Autoencoders employing different NN types (Kramer 1991; Murata et al. 2020; Xu and Duraisamy 2020; Maulik et al. 2021) and other nonlinear extensions of the previous techniques are a widely used approach for dealing with nonlinear cases typical in fluid dynamics (Gonzalez and Balajewicz 2018). These techniques also frequently include physics information to guarantee consistency (Erichson et al. 2019).

1.4.3 Structural Mechanics Applications

ML has been used for some time already in structural mechanics, with probably the most applications in Structural Health Monitoring (SHM) (Farrar and Worden 2012). ML is applied for the primal identification of structural systems (SSI) (Sirca and Adeli 2012; Amezquita-Sancheza et al. 2020), in particular of complex or historical structures, to assess their general and seismic vulnerability (Ruggieri et al. 2021; Xie et al. 2020) and facilitate ulterior health monitoring (Mishra 2021). Feature extraction and model reduction is fundamental in these approaches (Rosafalco et al. 2021). Other areas where ML is employed is in the control of structures (e.g., active Tuned Mass Dampers, Yucel et al. 2019; Colherinhas et al. 2019; Etedali and Mollayi 2018) under wind, seismic or crowd actions, or in structural design (Herrada et al. 2017; Sun et al. 2021; Hong et al. 2020; Yuan et al. 2020). We also comment in this section on the development of novel ML approaches based on ideas used in structural and finite element analyses.

1.4.3.1 Structural System Identification and Health Monitoring

Structural System Identification (SSI) is a key in analyzing the vulnerability of historical structures in seismic zones (e.g., Italy and Spain) (Domaneschi et al. 2021). It is also a problem in the assessment of modern structures, since modeling assumptions may not have been sufficiently accurate (Torky and Ohno 2021). Many classical approaches based on optimization methods are frequently ill-conditioned or they present many possible solutions, some of which should have been discarded automatically. Hence, ML is an excellent approach to address SSI, and different algorithms have been employed. For example, SVM (Gui et al. 2017), and in particular Weighted Least Squares Support Vector Machines (LS-SVM) have been employed to determine the structural parameters and then identify degradation due to damage through dynamic response (Tang et al. 2006; Zhang et al. 2007). K-Means and KNNs are also frequently used in SHM. For example, in Sarmadi and Karamodin (2020) anomaly detection is performed using the (squared) “distance” $(\mathbf{x} - \bar{\mathbf{x}})\mathbf{S}_k^{-1}(\mathbf{x} - \bar{\mathbf{x}})$ to detect the k-NN in a multivariate one-class k-NN approach. The authors applied the approach to wood and steel bridges and compared the results obtained with other ML techniques to reach the smallest misclassification rate. Bridge structures have also been focused on using Genetic Algorithms in an unsupervised approach to detect damage (Silva et al. 2016). Health monitoring of bridges is the focus in rather early research (e.g., the simple case analyzed in Liu and Sun 1997 through NNs). The traffic load (Lee et al. 2002) and ambient vibrations (Avci et al. 2021) are often actions that require the study of the evolution of the mechanical properties. The application of NNs is typical in detecting changes in the properties and the possible explanations on the origin of those changes (Ko and Ni 2005). Basically, all types of bridges have been studied using ML techniques, namely steel girder bridges (Nick et al. 2021), reinforced concrete T-bridges (Hasançebi and Dumlupınar 2013), cable stayed (Arangio and Bontempi 2015) and long suspension bridges (Ni et al. 2020),

truss bridges (Mehrzoo et al. 2008), and arch bridges (Jayasundara et al. 2020). Different types of NN are used (e.g., Bayesian, Arangio and Beck 2012; Li et al. 2020; Ni et al. 2001, Convolutional, Nguyen et al. 2020; Quqa et al. 2022, Recurrent, Miao et al. 2023; Miao and Yokota 2022), and the use of other techniques is also frequent as SVM; see, for example, (Alamdari et al. 2017; Yu et al. 2021).

Apart from bridges and multi-story buildings (González and Zapico 2008; Wang et al. 2020), there are many other types of structures for which SSI and SHM are performed employing ML. Important structures are dams, where a deterioration and failure may cause massive destruction, hence visual inspection and monitoring of displacement cycles are typical actions in SHM of dams. The observations feed ML algorithms to assess the health of the structure. The estimation of the structural response from collected data is studied for example in Li et al. (2021b), where a CNN is used to extract features and a bidirectional gated RNN is employed to perform transfer learning from long-term dependencies. Similar works addressing SHM of dams are given (Yuan et al. 2022; Sevieri and De Falco 2020). A review may be found in Salazar et al. (2017).

Of course, different outputs may be pursued and the appropriate ML technique is related to both available data and desired output. For example, NNs have been used in Kao and Loh (2013), Ranković et al. (2012), Chen et al. (2018), and He et al. (2022) to monitor radial and lateral displacements in arch dams. Several ML techniques such as Random Forest (RF), Boosted Regression Trees (BRT), NN, SVM, and MARS are compared in Salazar et al. (2015) in the prediction of dam displacements and of dam leakage. The researchers found that BRT outperforms the most common data-driven technique employed when considering this problem, namely the Hydrostatic-Seasonal-Time method (HST), which accounts for the irreversible evolution of the dam response due to the reversible hydrostatic and thermal loads; see also Salazar et al. (2016). Gravity dams are a different type of structure from arch dams. Their reliability under flooding, earthquakes, and aging has also been addressed using ML methods in Hariri-Ardebili and Pourkamali-Anaraki (2018), where kNN, SVM, and NB have been used in the binary classification of structural results, and a failure surface is computed as a function of the dimensions of the dam. Related to dam infrastructure planning, flooding susceptibility predictions due to rainfall using NB and Naïve Bayes Trees (NBT) are compared in Khosravi et al. (2019) with three classical methods (see review of Multicriteria Decision Making (MCDM) in de Brito and Evers 2016) in the field. For tunnel designs and monitoring, we have that the soil is also an integral part of the structure and is difficult to characterize. The understanding of its behavior often depends on qualitative observations; it is therefore another field where machine learning techniques will have an important impact in the future (Jafari 2020).

Important types of structures considered in SHM are also aerogenerators or Wind Turbines (WT); see review in Ciang et al. (2008). Here, two main components are typically analyzed: the blades and the gearbox (Wang et al. 2016). SVM is a frequent ML technique used and acoustic noise is a source of relevant data for blade monitoring (Regan et al. 2016). Deep NNs are also frequently employed when multiple sources of data are available, in particular CNNs are used to deal with images from drones (Shihavuddin et al. 2019; Guo et al. 2021). Images are valuable not only in the detection of overall damage (e.g., determining a damage index value), but also in

determining the location of the damage. This gives an alternative to the placement of networks of strain sensors (Laflamme et al. 2016). Other WT functional problems, such as dirt and mud detection in blades to improve maintenance, can be determined employing different ML methods; e.g., in Jiménez et al. (2020) k -Nearest Neighbors (k -NN), SVM, LDA, PCA, DT, and an ensemble subspace discriminant method are employed. Other factors like the presence of ice in cold climates are also important. In Jiménez et al. (2019), a ML approach is applied to pattern recognition on guided ultrasonic waves to detect and classify ice thickness. In this work, different ML techniques are employed for feature extraction (data reduction into meaningful features), both linear (autoregressive ML models and PCA) and nonlinear (nonlinear-AR exogenous and nonlinear PCA), and then feature selection is performed to avoid overfitting. A wide range of supervised classifiers of different families (DT, LDA, QDA, several types of SVM, k NN, and ensembles) were employed and compared, both in terms of accuracy and efficiency.

Applications of ML can be found also in data preparation, including imputation techniques to fill missing sensor data (Li et al. 2021a, b). Systems, and damage and structural responses are assessed employing different variables. Typical variables are the displacements (building drift), which allow for the determination of material and structural geometric properties, for example in reinforced concrete (RC) columns. This can be achieved through locally weighted LS-SVM (Luo and Paal 2019). Bearing capacities and failure modes of structural components (columns, beams, shear walls) can also be predicted using ML techniques, in particular when the classical methods are complex and lack accuracy. For example, in Mangalathu et al. (2020) several ML methods such as Naïve Bayes, k NN, decision trees, and random forests combined with several weighted Boost techniques (similar to ensemble learning under the assumption that many weak learners make a strong learner) such as AdaBoost (Adaptive Boost, meaning that new weak learners adapt from misclassifications of previous ones) are compared to predict the failure modes (flexural, diagonal tension or compression, sliding shear) of RC shear walls in seismic events.

Identification of smart structures with nonlinearities, like buildings with magnetorheological dampers, has been performed through a combination of NN, PCA, and fuzzy logic (Mohammadzadeh et al. 2015).

In SHM, the integration of data from different types or families of sensors (data fusion) is an important topic. Data fusion (Hall and Llinas 2001) brings not only challenges in SHM but also the possibility of more accurate, integral prediction of the health of the structure (Wu and Jahanshahi 2020). For example, in Vitola et al. (2017) a data fusion system based on k NN classification was used in SHM. SHM is most frequently performed through the analysis of the dynamic response of the structure and comparing vibrational modes using the Modal Assurance Criterion (MAC) (Ho et al. 2021). However, in the more challenging SSI, many other additional features are employed as typology, age, and images. In SHM, damage detection is also pursued through the analysis of images. Visual inspection is a long used method for crack detection in concrete or steel structures, or to determine unusual displacements and deformations of the overall structure from global structural images. Automatic processing and damage detection from images obtained from stationary cameras or an

Unmanned Aerial Vehicle (UAV) (Sankarasrinivasan et al. 2015; Reagan et al. 2018) is currently being performed using ML techniques. A recent review of these image-based techniques can be found in Dong and Catbas (2021). Another recent review of ML applications of SHM of civil structures can be found in Flah et al. (2021).

One of the lessons learnt considering the available results is that to improve predictions and robustness, some progress is needed in physics-based ML approaches for SHM. For instance, an improvement may be using concrete damage models with environmental data, typology, images, etc. to detect damage which may have little impact in sensors (Kralovec and Schagerl 2020), but which may result in significant losses. This issue is also of special relevancy in the aircraft industry (Ahmed et al. 2021).

1.4.3.2 Structural Design and Topology Optimization

The design of components and structures is based on creativity and experience (Málaga-Chuquitaype 2022), so it is also an optimal field for the use of ML procedures, e.g., Adeli and Yeh (1990). ML in the general design of industrial components is briefly addressed below.

Given the creative nature of structural design, evolutionary algorithms are good choices. For example in Freischlad and Schnellenbach-Held (2005), linguistic modeling is applied to conceptual design, investigating evolutionary design and optimization of high-rise concrete buildings for lateral load bearing. The process of the design of a structure using ML from concept to actual structural detailing is discussed in Chang and Cheng (2020). Different structural systems are conceptually designed with the aid of ML techniques, including shear walls to sustain earthquakes (e.g., using GAN, in Lu et al. 2022; Zhao et al. 2022), shell structures (Tam et al. 2020; Zheng et al. 2020), and even the architectural volume (Chang et al. 2021). A study and proposal of different ML techniques in architectural design can be found in Tamke et al. (2018).

Of course, one of the main disciplines in structural design is Topology Optimization (TO), and ML approaches (a combination coined “learning topology” in Moroni and Pascali 2021) can be used to develop more robust schemes (Chi et al. 2021; Muñoz et al. 2022) through tuning numerical parameters (Lynch et al. 2019). For example in Muñoz et al. (2022), manifold learning approaches such as local linear embedding (LLE) techniques are employed to extract geometrical modes defined by the material distribution given by the TO algorithm, facilitating the creation of new geometries. TO of nonlinear structures is also performed using deep NN (Abueidda et al. 2020). In order to obtain optimum thermal structures, GANs have been used to develop non-iterative structural to Li et al. (2019). Using ML to develop a non-iterative TO approach has also been addressed in Yu et al. (2019). A recent review of ML techniques in TO can be found in Mukherjee et al. (2021).

1.4.4 Machine Learning Approaches Motivated in Structural Mechanics and by Finite Element Concepts

While ML has contributed to CAE and structural design, new ML approaches have also been developed based on concepts that are traditional in structural analysis and finite element solutions. For example, one of the ideas is the concept of substructuring, employed in static condensation, Guyan reduction, and Craig–Bampton schemes (Bathe 2006). In Jokar and Semperlotti (2021) a Finite Element Network Analysis (FENA) is proposed. The method substitutes the classical finite elements by a library of “elements” consisting of a Bidirectional Recurrent Neural Network (BRNN). The BRNN of the elements are trained individually and the training can be computationally costly. Then these trained BRNN are concatenated, and the composite system needs no further training. The solution is fast, not considering the training, since in contrast to FE solutions, no system of equations is solved. The method has only been applied to the analysis of an elastic bar, so the generalization of the idea to the solution of more complex problems is still an open research task.

The partition of unity used in finite element and meshless methods has been employed to develop a Finite Element Machine (FEMa) for fast supervised learning (Pereira et al. 2020). The idea is that each training sample is the center of a Shepard function, and the training set is treated as a probabilistic manifold. The advantage is that, as in the case of spline-based approaches, the technique has no parameters. Compared to several methods, the BPNN, Naïve Bayes, SVM (using both RBF and sigmoids), RF, DT, etc. the FEMa method was competitive in the eighteen benchmark datasets typically employed in the literature when analyzing supervised methods.

Another interesting approach is the substitution of some procedures of finite element methods with machine learning approaches. Candidates are material and general element libraries, creating surrogate material models (discussed above) or surrogate elements, or patches of elements. This approach follows the substructuring or multiscale computational homogenization (FE2) idea, but in this case using ML procedures instead of a RVE finite element mesh. In Capuano and Rimoli (2019), several possibilities are addressed and applied to nonlinear truss structures and a (nonlinear) hyperelastic perforated plane strain structure. A similar approach is used in Yan et al. (2022) for composite shells employing physics-based NNs. In Jung et al. (2020), finite element matrices passing the patch test are generated from data using a neural network accounting for some physical constraints, as vanishing strain energy under rigid body motions.

1.4.5 Multiphysics Problems

Despite the already mentioned advance in scientific machine learning in several fields, much less has been achieved considering multiphysics problems. This is undoubtedly due to the youth of the discipline, but there are a number of efforts that deserve

mentioning. For instance, in Alexiadis (2019) a system is developed with the aim of replicating human physiology. In Alizadeh et al. (2021), a similar approach is developed for nanofluid flow, while (Ren et al. 2020) studies hydrogen production.

In the field of multiphysics problems, there exists a particularly appealing approach to machine learning, namely that of port-Hamiltonian formalisms (Van Der Schaft et al. 2014). Port-Hamiltonian systems are essentially open systems that obey a Hamiltonian description of their physics (and thus, are conservative, or reversible). Their interaction with the environment is made through a forcing term. If we call \mathbf{z} the set of variables governing the problem ($\mathbf{z} = (\mathbf{p}, \mathbf{q})$, e.g., position and momentum, for a canonical Hamiltonian system), its evolution in time will be given by

$$\dot{\mathbf{z}} = \mathbf{J}\nabla H + \mathbf{F} \quad (1.61)$$

where \mathbf{J} is the classical (skew-symmetric) symplectic matrix, H is the Hamiltonian (total energy of the system), and \mathbf{F} is the forcing term, which links the port-Hamiltonian system to other subsystems. This paves the way for an efficient coupling of different systems, possibly governed by different physics. Enforcing this port-Hamiltonian structure during the learning process, as an inductive bias, ensures the fulfillment of the conservation of energy in the total system, while allowing for a proper introduction of dissipative terms in the formulation. This is indeed the approach followed in Desai et al. (2021); see also Massaroli et al. (2019), Eidnes et al. (2022), Mattheakis et al. (2022), Sprangers et al. (2014), and Morandin et al. (2022). A recent review on the progress of these techniques can be found in Cherifi (2020).

1.4.6 Machine Learning in Manufacturing and Design

ML techniques have been applied to classical manufacturing since their early conception, and are now important in Additive Manufacturing (AM). Furthermore, ML is currently being applied to the complete product chain, from conceptual design to the manufacturing process. Below, we review ML applications in classical and additive manufacturing, and in automated design.

1.4.6.1 Classical Manufacturing

In manufacturing, plasticity plays a fundamental role. Machine learning approaches to solve inelastic problems have already been addressed in Sect. 1.4.1.2 above. Research activities prior to 1965 are compiled in an interesting review by Monostori et al. (1996). More recently, another review compiled the works in different research endeavors within the field of manufacturing (Pham and Afify 2005).

Of course, ML is a natural ally of the Industry 4.0 paradigm (the fourth industrial revolution), in which sensors are ubiquitous and data streams provide the systems with valuable information. This synergistic alliance is explored in Raj et al. (2021). In Sharp et al. (2018), valuable research is reported in which Natural Language Pro-

cessing (NLP) was applied to documentation from 2005 to 2017 in the field of smart manufacturing. The survey analyzes aspects ranging from decision support (prior to the moment, a piece was manufactured), plant and operations health management (for the manufacturing process itself), data management, as a consequence of the vast amount of information produced by Internet of Things (IoT) devices installed in modern plants, or lifecycle management, for instance. The survey concludes that ML-based techniques are present in the literature (at the moment of publication, 2018) for product life cycle management. While many of these ML techniques are inherently designed to perform prognosis (i.e., to predict several aspects related to manufacturing), in Ademuji et al. (2017) a review is given of literature that employs ML to perform diagnosis of manufacturing processes.

1.4.6.2 Additive Manufacturing

Due to its inherent technological complexity and our still limited comprehension of many of the physical processes taking place, additive manufacturing (AM) has been an active field of research in machine learning. The interested reader can consult different reviews of the state of the art (Razvi et al. 2019; Meng et al. 2020; Jin et al. 2020; Wang et al. 2020). One of the fields where ML will be very important, and that is tied to topology optimization, is 3D printing. AM, in particular 3D printing, represents a revolution in component design and manufacturing because it allows for infinite possibilities and largely reduced manufacturing difficulties. Moreover, these technologies are reaching resolutions at the microscale, so a component may be designed and manufactured with differently designed structures at the mesoscale (establishing metamaterials), obtaining unprecedented material properties at the continuum scale thus widening the design space (Barchiesi et al. 2019; Zadpoor 2016).

There are many different AM procedures, like Fused Deposition Modeling (FDM), Selective Laser Melting (SLM), Direct Energy Deposition (DED), Electron Beam Melting (EBM), Binder Jetting, etc. While additive manufacturing offers huge possibilities, it also results into associated new challenges in multiple aspects, from the detection of porosity (important in the characterization of the printed material) to the recognition of defects (melting, microstructural, and geometrical), to the characterization of the complex anisotropic behavior, which depends on multiple parameters of the manufacturing process (e.g., laser power in Selected Laser Melting, direction of printing, powder and printing conditions). Both the design using AM and the error correction or compensation (Omairi and Ismail 2021) are typical objectives in the application of ML to AM. Different ML techniques are employed, with SVM one of the most used schemes. For example, SVM is employed for identifying defective parts from images in FDM (Delli and Chang 2018), for detecting geometrical defects in SLM-made components (zur Jacobsmühlen et al. 2015; Gobert et al. 2018), for building process maps relating variables to desired properties (e.g., as low porosity) (Aoyagi et al. 2019), and for predicting surface roughness in terms of process features (Wu et al. 2018). NNs are often used for optimizing the AM process by predicting properties as a function of printing variables. For example, NNs have been used for

predicting and optimizing melt pool geometry in DED (Caiazzo and Caggiano 2020), to build process maps and optimize efficiency and surface roughness in SLM (Zhang et al. 2017), to minimize support wasted material (optimize supports in a piece) in FDM (Jiang et al. 2019), to predict and optimize resulting mechanical properties of the printed material (Lewandowski and Seifi 2016) like strength (e.g., using CNN from thermal histories in Xie et al. 2021 or FFNN in Bayraktar et al. 2017), bending stiffness in AM composites (Nawafleh and AL-Oqla 2022), and stress–strain curves of binary composites using a combination of CNN and PCA (Yang et al. 2020).

NNs have also been used to create surrogate models with the purpose of mimicking the acoustic properties of AM replicas of a Stradivarius violin (Tian et al. 2021). Reviews of techniques and different applications of machine learning in additive manufacturing may be found in Wang et al. (2020), DebRoy et al. (2021), Meng et al. (2020), Qin et al. (2022), Xames et al. (2023), and Hashemi et al. (2022). The review in Guo et al. (2022) addresses in some detail physics-based proposals.

1.4.6.3 Automated CAD and Generative Design

A fundamental step in the design of an industrial component or an architected structure is the conceptual development of the novel component or structure (the most creative part), and more often, the customization of a component from a given family to meet the specific requirements of the component in the system to which it will be added. The novel product is in essence a variation or evolution of previous concepts (first case) or previous components (second case). ML may help in both cases. The challenge of understanding the “rules” of creativity to foster it has paved the way for interesting contributions of ML in this field (Ganin et al. 2021).

In the first case, ML helps in the generative design of a novel component or structure by creating variations supported by attributes, based in essence on the combination and evolution of previous conceptual designs (Gero 1996; Khan and Awan 2018); see the review of ML contributions in Duffy (1997); see also Tzonis and White (2012) especially for conceptual design. An example would be to create a new design of a car. Some conditions are given by the segment to which it will belong, but some other possibilities are open and can be generated from possible variations that may please or attract consumers. For example, Generative Adversarial Networks (GAN) (Goodfellow et al. 2020) are used to explore aerodynamic shapes (Chen et al. 2019). ML is also used for the association of concepts and combinatorial creativity with aims to the reuse of creativity to create new concepts and designs (Chen 2020). Further, ML is employed in the evaluation of design concepts from many candidates based on human preferences expressed in previous concepts (Camburn et al. 2020). There are also ML works that aid in the development of detailed and consistent CAD drawings from hand sketches (Seff et al. 2021), i.e. interpreting and detailing a CAD drawing from a hand sketch.

Considering the second case, the customization of designs is natural to ML approaches. The idea here is to perform automatic variations of previous conceptual designs, or of designs obtained from mathematical optimization. A good example

of this approach using deep NN is given in Yoo et al. (2021), to propose designs of a wheel, in which variations that comply with mechanical requirements (strength, eigenfrequencies, etc. evaluated through surrogate models as a function of geometric parameters) given with shapes are obtained by variations and simplifications using autoencoders. Based on this work, an interesting discussion between aesthetics and performance (aspects to include in ML models) is given in Shin et al. (2021). The combination of topology optimization and generative design can be found in many endeavors (Oh et al. 2019; Barbieri and Muzzupappa 2022).

Moreover, in the design process, there are many aspects that can be automated. A typical aspect is the search for components with similar layout such that detailed drawings, solid models (Chu and Hsu 2006), and manufacturing processes (Li et al. 2016) of new designs may be inferred from previous similar designs (Zehtaban et al. 2016). Indeed, many works focus on procedures to reuse parts of CAD schemes for electronic circuits (Boning et al. 2019) or to develop microfluidic devices (Lore et al. 2015; Tsur 2020).

1.5 Conclusions

With the current access to large amounts of data and the ubiquitous presence of real-time sensors in our life, as those present in cell phones, and also with the increased computational power, Machine Learning (ML) has resulted in a change of paradigm on how many problems are addressed. When using ML, the approach to many engineering problems is no longer a matter of understanding the governing equations, not even a matter of fully understanding the problem being addressed, but of having sufficient data so relations between features and desired outputs can be established; and not even in a deterministic way, but in an implicit probabilistic way.

ML has been succeeding for more than a decade in solving complex problems as face recognition or stocks evolution, for which there was no successful deterministic method, and not even a sound understanding of the actual significance of the main variables affecting the result. Computer Aided Engineering (CAE), with the Finite Element Method standing out, had also an extraordinary success in accurately solving complex engineering problems, but a detailed understanding of the governing equations and their discretization is needed. This success delayed the introduction of ML techniques in classical CAE dominated fields, but during the last years increasing emphasis has been placed on ML methods. In particular, ML is used to solve some of the issues still remaining when addressing the problem through classical techniques. Examples of these issues are the still limited generality of classical CAE methods (although the success of the FEM is due to its good generalization possibilities), the search for practical solutions when there is not a complete, full understanding of the problem, and computational efficiency in high-dimensional problems like multiscale and nonlinear inverse problems. While we are still seeing the start of a new era, already a large variety of problems in CAE has been addressed using different ML techniques.

Lessons have also been learned in the last few years. One important lesson is that in engineering solutions, robustness and reliability of the solution are important (Bathe 2006), and data may not be sufficient to guarantee that robustness. Then, ML methods that incorporate physical laws and use the vast analytical knowledge acquired in the last centuries may result not only in more robust methods but also in more efficient schemes. In this chapter, we briefly reviewed ML techniques in CAE and some representative applications. We focused on conveying some of the excitement that is now developing in the research and use of ML techniques by short descriptions of methods and many references to applications of those techniques.



Acknowledgements This is part of the training activities of the project funded by European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie Grant Agreement No. 101007815; FJM gratefully acknowledges this funding.

The Support of the Spanish Ministry of Science and Innovation, AEI/10.13039/501100011033, through Grant number PID2020-113463RB-C31 and by the Regional Government of Aragon, grant T24-20R, and the European Social Fund is also gratefully acknowledged by EC.

References

- Abdar M, Pourpanah F, Hussain S, Rezazadegan D, Liu L, Ghavamzadeh M, Fieguth P, Cao X, Khosravi A, Acharya UR et al (2021) A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Inf Fusion* 76:243–297
- Abdi H, Williams LJ (2010) Principal component analysis. *Wiley Interdiscip Rev: Comput Stat* 2(4):433–459
- Abueidda DW, Koric S, Sobh NA (2020) Topology optimization of 2d structures with nonlinearities using deep learning. *Comput Struct* 237:106283
- Abueidda DW, Koric S, Sobh NA, Sehitoglu H (2021) Deep learning for plasticity and thermo-viscoplasticity. *Int J Plast* 136:102852
- Adeli H, Yeh C (1990) Explanation-based machine learning in engineering design. *Eng Appl Artif Intell* 3(2):127–137
- Ademuji TT, Brundage MP, Prabhu VV (2017) A review of current machine learning techniques used in manufacturing diagnosis. In: IFIP international conference on advances in production management systems. Springer, Berlin, pp 407–415
- Adriaans P, Zantinge D (1997) Data mining. Addison-Wesley Longman Publishing Co., Inc
- Aggarwal A, Mittal M, Battineni G (2021) Generative adversarial network: an overview of theory and applications. *Int J Inf Manag Data Insights* 1(1):100004
- Agarwal A, Choudhary A (2016) Perspective: materials informatics and big data: realization of the “fourth paradigm” of science in materials science. *Apl Mater* 4(5):053208
- Ahmed O, Wang X, Tran MV, Ismadi MZ (2021) Advancements in fiber-reinforced polymer composite materials damage detection methods: towards achieving energy-efficient shm systems. *Compos Part B: Eng* 223:109136
- Ahmed E, Jones M, Marks TK (2015) An improved deep learning architecture for person re-identification. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3908–3916
- Akinyelu AA (2021) Advances in spam detection for email spam, web spam, social network spam, and review spam: ML-based and nature-inspired-based techniques. *J Comput Secur* 29(5):473–529

- Alamdari MM, Rakotoarivelo T, Khoa NLD (2017) A spectral-based clustering for structural health monitoring of the sydney harbour bridge. *Mech Syst Signal Process* 87:384–400
- Alber M, Buganza Tepole A, Cannon WR, De S, Dura-Bernal S, Garikipati K, Karniadakis G, Lytton WW, Perdikaris P, Petzold L et al (2019) Integrating machine learning and multiscale modeling-perspectives, challenges, and opportunities in the biological, biomedical, and behavioral sciences. *NPJ Digit Med* 2(1):1–11
- Alletti M, Bortolossi A, Perotto S, Veneziani A (2015) One-dimensional surrogate models for advection-diffusion problems. In: *Numerical mathematics and advanced applications-ENUMATH 2013*. Springer, Berlin, pp 447–455
- Alexiadis A (2019) Deep multiphysics: Coupling discrete multiphysics with machine learning to attain self-learning in-silico models replicating human physiology. *Artif Intell Med* 98:27–34
- Alizadeh R, Allen JK, Mistree F (2020) Managing computational complexity using surrogate models: a critical review. *Res Eng Des* 31(3):275–298
- Alizadeh R, Abad JMN, Ameri A, Mohebbi MR, Mehdizadeh A, Zhao D, Karimi N (2021) A machine learning approach to the prediction of transport and thermodynamic processes in multiphysics systems-heat transfer in a hybrid nanofluid flow in porous media. *J Taiwan Inst Chem Eng* 124:290–306
- Alvarez MA, Rosasco L, Lawrence ND et al (2012) Kernels for vector-valued functions: A review. *Found Trends® Mach Learn* 4(3):195–266
- Amezquita-Sancheza J, Valtierra-Rodríguez M, Adeli H (2020) Machine learning in structural engineering. *Scientia Iranica* 27(6):2645–2656
- Amores VJ, Benítez JM, Montáns FJ (2020) Data-driven, structure-based hyperelastic manifolds: A macro-micro-macro approach to reverse-engineer the chain behavior and perform efficient simulations of polymers. *Comput Struct* 231:106209
- Amores VJ, Nguyen K, Montáns FJ (2021) On the network orientational affinity assumption in polymers and the micro-macro connection through the chain stretch. *J Mech Phys Solids* 148:104279
- Amores VJ, San Millán FJ, Ben-Yelun I, Montáns FJ (2021) A finite strain non-parametric hyperelastic extension of the classical phenomenological theory for orthotropic compressible composites. *Compos Part B: Eng* 212:108591
- Angelov S, Stoimenova E (2017) Cross-validated sequentially constructed multiple regression. In: *Annual meeting of the bulgarian section of SIAM*. Springer, Berlin, pp 13–22
- Aoyagi K, Wang H, Sudo H, Chiba A (2019) Simple method to construct process maps for additive manufacturing using a support vector machine. *Addit Manuf* 27:353–362
- Arangio S, Beck J (2012) Bayesian neural networks for bridge integrity assessment. *Struct Control Health Monit* 19(1):3–21
- Arangio S, Bontempi F (2015) Structural health monitoring of a cable-stayed bridge with bayesian neural networks. *Struct Infrastruct Eng* 11(4):575–587
- Arbabi H, Bunder JE, Samaey G, Roberts AJ, Kevrekidis IG (2020) Linking machine learning with multiscale numerics: Data-driven discovery of homogenized equations. *JOM* 72(12):4444–4457
- Arrieta AB, Díaz-Rodríguez N, Del Ser J, Bennetot A, Tabik S, Barbado A, García S, Gil-López S, Molina D, Benjamins R et al (2020) Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Inf Fusion* 58:82–115
- Asgari S, MirhoseiniNejad S, Moazamigoodarzi H, Gupta R, Zheng R, Puri IK (2021) A gray-box model for real-time transient temperature predictions in data centers. *Appl Therm Eng* 185:116319
- Ashtiani MN, Raahemi B (2021) Intelligent fraud detection in financial statements using machine learning and data mining: a systematic literature review. *IEEE Access* 10:72504–72525
- Aubry N (1991) On the hidden beauty of the proper orthogonal decomposition. *Theor Comput Fluid Dyn* 2(5):339–352
- Audouze C, De Vuyst F, Nair P (2009) Reduced-order modeling of parameterized PDEs using time-space-parameter principal component analysis. *Int J Numer Methods Eng* 80(8):1025–1057

- Avcı O, Abdeljaber O, Kiranyaz S, Hussein M, Gabbouj M, Inman DJ (2021) A review of vibration-based damage detection in civil structures: From traditional methods to machine learning and deep learning applications. *Mech Syst Signal Process* 147:107077
- Ayensa Jiménez J (2022) Study of the effect of the tumour microenvironment on cell response using a combined simulation and machine learning approach. application to the evolution of Glioblastoma. Ph.D. thesis, School of Engineering and Architecture. Universidad de Zaragoza
- Baklacioglu T, Turan O, Aydin H (2019) Metaheuristics optimized machine learning modelling of environmental exergo-emissions for an aero-engine. *Int J Turbo Jet-Engines* 39(3):411–426
- Balakrishnama S, Ganapathiraju A (1998) Linear discriminant analysis-a brief tutorial. *Inst Signal Inf Process* 18:1–8
- Bank D, Koenigstein N, Giryas R (2020) Autoencoders. [arXiv:2003.05991](https://arxiv.org/abs/2003.05991)
- Barbieri L, Muzzupappa M (2022) Performance-driven engineering design approaches based on generative design and topology optimization tools: A comparative study. *Appl Sci* 12(4):2106
- Barchiesi E, Spagnuolo M, Placidi L (2019) Mechanical metamaterials: a state of the art. *Math Mech Solids* 24(1):212–234
- Bastek JH, Kumar S, Telgen B, Glaesener RN, Kochmann DM (2022) Inverting the structure–property map of truss metamaterials by deep learning. *Proc Natl Acad Sci* 119(1) (2022)
- Bathe KJ (2006) *Finite element procedures*, 2nd edn 2014, KJ Bathe, Watertown, MA; also published by Higher Education Press China 2016
- Bathe KJ, Wilson EL (1973) Solution methods for eigenvalue problems in structural mechanics. *Int J Numer Methods Eng* 6(2):213–226
- Bayraktar Ö, Uzun G, Çakiroğlu R, Guldaz A (2017) Experimental study on the 3d-printed plastic parts and predicting the mechanical properties using artificial neural networks. *Polym Adv Technol* 28(8):1044–1051
- Becker M, Teschner M (2007) Weakly compressible SPH for free surface flows. In: *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on computer animation*, pp. 209–217
- Beel J, Gipp B (2009) Google scholar’s ranking algorithm: an introductory overview. In: *Proceedings of the 12th international conference on scientometrics and informetrics (ISSI’09)*, vol 1. Rio de Janeiro (Brazil), pp 230–241
- Belkin M, Niyogi P (2003) Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput* 15(6):1373–1396
- Bengio Y, Courville A, Vincent P (2013) Representation learning: A review and new perspectives. *IEEE Trans Pattern Anal Mach Intell* 35(8):1798–1828
- Benítez JM, Montáns FJ (2018) A simple and efficient numerical procedure to compute the inverse langevin function with high accuracy. *J Non-Newton Fluid Mech* 261:153–163
- Berkooz G, Holmes P, Lumley JL (1993) The proper orthogonal decomposition in the analysis of turbulent flows. *Annu Rev Fluid Mech* 25(1):539–575
- Bertoldi K, Vitelli V, Christensen J, Van Hecke M (2017) Flexible mechanical metamaterials. *Nat Rev Mater* 2(11):1–11
- Bezazi A, Pierce SG, Worden K et al (2007) Fatigue life prediction of sandwich composite materials under flexural tests using a bayesian trained artificial neural network. *Int J Fatigue* 29(4):738–747
- Bhatti G, Mohan H, Singh RR (2021) Towards the future of smart electric vehicles: Digital twin technology. *Renew Sustain Energy Rev* 141:110801
- Bickel S, Haider P, Scheffer T (2005) Learning to complete sentences. In: *European conference on machine learning*. Springer, Berlin, pp 497–504
- Bird GD, Gorrell SE, Salmon JL (2021) Dimensionality-reduction-based surrogate models for real-time design space exploration of a jet engine compressor blade. *Aerosp Sci Technol* 118:107077
- Bischl B, Lang M, Kotthoff L, Schiffner J, Richter J, Studerus E, Casalicchio G, Jones ZM (2016) mlr: Machine learning in R. *J Mach Learn Res* 17(1):5938–5942
- Bishop CM (1995) Training with noise is equivalent to tikhonov regularization. *Neural Comput* 7(1):108–116
- Bisong E (2019a) Google cloud machine learning engine (cloud MLE). In: *Building machine learning and deep learning models on google cloud platform*, pp. 545–579. Springer, Berlin

- Bisong E (2019b) Numpy. In: Building machine learning and deep learning models on google cloud platform. Springer, Berlin, pp 91–113
- Bock FE, Aydin RC, Cyron CJ, Huber N, Kalidindi SR, Klusemann B (2019) A review of the application of machine learning and data mining approaches in continuum materials mechanics. *Frontiers Mater* 6:110
- Bonet J, Wood RD (1997) *Nonlinear continuum mechanics for finite element analysis*. Cambridge University Press
- Boning DS, Elfadel IAM, Li X (2019) A preliminary taxonomy for machine learning in vlsi cad. In: *Machine learning in VLSI computer-aided design*. Springer, Berlin, pp 1–16
- Borkowski L, Sorini C, Chattopadhyay A (2022) Recurrent neural network-based multiaxial plasticity model with regularization for physics-informed constraints. *Comput Struct* 258:106678
- Braconnier T, Ferrier M, Jouhaud JC, Montagnac M, Sagaut P (2011) Towards an adaptive pod/svd surrogate model for aeronautic design. *Comput Fluids* 40(1):195–209
- Bro R, Smilde AK (2014) Principal component analysis. *Anal Methods* 6(9):2812–2831
- Brodie CR, Constantin A, Deen R, Lukas A (2020) Machine learning line bundle cohomology. *Fortschritte der Physik* 68(1):1900087
- Brunton SL, Kutz JN (2022) *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press (2022)
- Brunton SL, Kutz JN (2019) Methods for data-driven multiscale model discovery for materials. *J Phys: Mater* 2(4):044002
- Brunton SL, Proctor JL, Kutz JN (2016) Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proc Natl Acad Sci* 113(15):3932–3937
- Build powerful models. (2022). <https://www.opennn.org/>
- Buizza C, Casas CQ, Nadler P, Mack J, Marrone S, Titus Z, Le Cornec C, Heylen E, Dur T, Ruiz LB et al (2022) Data learning: integrating data assimilation and machine learning. *J Comput Sci* 58:101525
- Bukka SR, Magee AR, Jaiman RK (2020) Deep convolutional recurrent autoencoders for flow field prediction. In: *International conference on offshore mechanics and arctic engineering*, vol 84409. American Society of Mechanical Engineers, p V008T08A005
- Burkov A (2020) *Machine learning engineering*, vol 1. True Positive Incorporated
- Burkov A (2019) *The hundred-page machine learning book*, vol 1. Andriy Burkov Quebec City, QC, Canada
- Burov A, Burova O (2020) Development of digital twin for composite pressure vessel. *J Phys: Conf Ser* 1441:012133. IOP Publishing
- Buşoni L, de Bruin T, Tolić D, Kober J, Palunko I (2018) Reinforcement learning for control: Performance, stability, and deep approximators. *Ann Rev Control* 46:8–28
- Bzdok D, Altman N, Krzywinski M (2018) Statistics versus machine learning. *Nat Methods* 15:233–234
- Caccin M, Li Z, Kermode JR, De Vita A (2015) A framework for machine-learning-augmented multiscale atomistic simulations on parallel supercomputers. *Int J Quantum Chem* 115(16):1129–1139
- Caiazza F, Caggiano A (2020) Laser direct metal deposition of 2024 Al alloy: trace geometry prediction via machine learning. *Materials* 11(3):444
- Camburn B, He Y, Raviselvam S, Luo J, Wood K (2020) Machine learning-based design concept evaluation. *J Mech Des* 142(3):031113
- Capuano G, Rimoli JJ (2019) Smart finite elements: A novel machine learning application. *Comput Methods Appl Mech Eng* 345:363–381
- Carleo G, Cirac I, Cranmer K, Daudet L, Schuld M, Tishby N, Vogt-Maranto L, Zdeborová L (2019) Machine learning and the physical sciences. *Rev Modern Phys* 91(4):045002
- Carrara P, De Lorenzis L, Stainier L, Ortiz M (2020) Data-driven fracture mechanics. *Comput Methods Appl Mech Eng* 372:113390
- Cayton L (2005) Algorithms for manifold learning. *Univ Calif San Diego Tech Rep* 12(1–17):1

- Champaney V, Chinesta F, Cueto E (2022) Engineering empowered by physics-based and data-driven hybrid models: A methodological overview. *Int J Mater Form* 15(3):1–14
- Champion K, Lusch B, Kutz JN, Brunton SL (2019) Data-driven discovery of coordinates and governing equations. *Proc Natl Acad Sci* 116(45):22445–22451
- Chan S, Elsheikh AH (2018) A machine learning approach for efficient uncertainty quantification using multiscale methods. *J Comput Phys* 354:493–511
- Chang KH, Cheng CY, Luo J, Murata S, Nourbakhsh M, Tsuji Y (2021) Building-gan: Graph-conditioned architectural volumetric design generation. In: *Proceedings of the IEEE/CVF international conference on computer vision*, pp 11956–11965
- Chang KH, Cheng CY (2020) Learning to simulate and design for structural engineering. In: *International conference on machine learning*. PMLR, pp 1426–1436
- Chattopadhyay A, Hassanzadeh P, Subramanian D (2020) Data-driven predictions of a multiscale Lorenz 96 chaotic system using machine-learning methods: reservoir computing, artificial neural network, and long short-term memory network. *Nonlinear Process Geophys* 27(3):373–389
- Chazal F, Michel B (2021) An introduction to topological data analysis: fundamental and practical aspects for data scientists. *Frontiers Artif Intell* 4:667363
- Chen L (2020) Data-driven and machine learning based design creativity. Ph.D. thesis, Imperial College London
- Chen T, Chen H (1995) Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE Trans Neural Netw* 6(4):911–917
- Chen W, Chiu K, Fuge M (2019) Aerodynamic design optimization and shape exploration using generative adversarial networks. In: *AIAA Scitech 2019 forum*, p 2351
- Chen S, Gu C, Lin C, Zhao E, Song J (2018) Safety monitoring model of a super-high concrete dam by using rbf neural network coupled with kernel principal component analysis. *Math Probl Eng* 1712653
- Cherifi K (2020) An overview on recent machine learning techniques for port hamiltonian systems. *Physica D: Nonlinear Phenomena* 411:132620
- Chi H, Zhang Y, Tang TLE, Mirabella L, Dalloro L, Song L, Paulino GH (2021) Universal machine learning for topology optimization. *Comput Methods Appl Mech Eng* 375:112739
- Chinesta F, Ladeveze P, Cueto E (2011) A short review on model order reduction based on proper generalized decomposition. *Arch Comput Methods Eng* 18(4):395–404
- Chinesta F, Cueto E, Abisset-Chavanne E, Duval JL, Khaldi FE (2020) Virtual, digital and hybrid twins: a new paradigm in data-based engineering and engineered data. *Arch Comput Methods Eng* 27(1):105–134
- Chinesta F, Cueto E, Grmela M, Moya B, Pavelka M, Šípka M (2020) Learning physics from data: a thermodynamic interpretation. In: *Workshop on joint structures and common foundations of statistical physics, information geometry and inference for learning*. Springer, Berlin, pp 276–297
- Choi SY, Cha D (2019) Unmanned aerial vehicles using machine learning for autonomous flight; state-of-the-art. *Adv Robot* 33(6):265–277
- Chu CH, Hsu YC (2006) Similarity assessment of 3d mechanical components for design reuse. *Robot Comput-Integr Manuf* 22(4):332–341
- Ciang CC, Lee JR, Bang HJ (2008) Structural health monitoring for a wind turbine system: a review of damage detection methods. *Meas Sci Technol* 19(12):122001
- Ciftci K, Hackl K (2022) Model-free data-driven simulation of inelastic materials using structured data sets, tangent space information and transition rules. *Comput Mech* 70:425–435
- Clifton A, Kilcher L, Lundquist J, Fleming P (2013) Using machine learning to predict wind turbine power output. *Environ Res Lett* 8(2):024009
- Coelho M, Roehl D, Bletzinger KU (2017) Material model based on NURBS response surfaces. *Appl Math Model* 51:574–586
- Colherinhas GB, de Moraes MV, Shzu MA, Avila SM (2019) Optimal pendulum tuned mass damper design applied to high towers using genetic algorithms: Two-dof modeling. *Int J Struct Stab Dyn* 19(10):1950125

- Conti S, Müller S, Ortiz M (2018) Data-driven problems in elasticity. *Arch Rat Mech Anal* 229(1):79–123
- Conti S, Müller S, Ortiz M (2020) Data-driven finite elasticity. *Arch Rat Mech Anal* 237(1):1–33
- Cranmer M, Greydanus S, Hoyer S, Battaglia P, Spergel D, Ho S (2020) Lagrangian neural networks. [arXiv:2003.04630](https://arxiv.org/abs/2003.04630)
- Crawford M, Khoshgoftaar TM, Prusa JD, Richter AN, Al Najada H (2015) Survey of review spam detection using machine learning techniques. *J Big Data* 2(1):1–24
- Crespo J, Montans FJ (2018) A continuum approach for the large strain finite element analysis of auxetic materials. *Int J Mech Sci* 135:441–457
- Crespo J, Montáns FJ (2019) General solution procedures to compute the stored energy density of conservative solids directly from experimental data. *Int J Eng Sci* 141:16–34
- Crespo J, Latorre M, Montáns FJ (2017) WYPIWYG hyperelasticity for isotropic, compressible materials. *Comput Mech* 59(1):73–92
- Crespo J, Duncan O, Alderson A, Montáns FJ (2020) Auxetic orthotropic materials: Numerical determination of a phenomenological spline-based stored density energy and its implementation for finite element analysis. *Comput Methods Appl Mech Eng* 371:113300
- de Brito MM, Evers M (2016) Multi-criteria decision-making for flood risk management: a survey of the current state of the art. *Nat Hazards Earth Syst Sci* 16(4):1019–1033
- De Jong K (1988) Learning with genetic algorithms: An overview. *Mach Learn* 3(2):121–138
- DebRoy T, Mukherjee T, Wei H, Elmer J, Milewski J (2021) Metallurgy, mechanistic models and machine learning in metal printing. *Nat Rev Mater* 6(1):48–68
- Delli U, Chang S (2018) Automated process monitoring in 3d printing using supervised machine learning. *Procedia Manuf* 26:865–870
- Demo N, Tezzele M, Rozza G (2018) Pydmd: Python dynamic mode decomposition. *J Open Source Softw* 3(22):530
- Dener A, Miller MA, Churchill RM, Munson T, Chang CS (2020) Training neural networks under physical constraints using a stochastic augmented Lagrangian approach. [arXiv:2009.07330](https://arxiv.org/abs/2009.07330)
- Deng Z, He C, Liu Y, Kim KC (2019) Super-resolution reconstruction of turbulent velocity fields using a generative adversarial network-based artificial intelligence framework. *Phys Fluids* 31(12):125111
- Denkena B, Bergmann B, Witt M (2019) Material identification based on machine-learning algorithms for hybrid workpieces during cylindrical operations. *J Intell Manuf* 30(6):2449–2456
- Desai SA, Mattheakis M, Sondak D, Protopapas P, Roberts SJ (2021) Port-hamiltonian neural networks for learning explicit time-dependent dynamical systems. *Phys Rev E* 104(3):034312
- Dhanalaxmi B (2020) Machine learning and its emergence in the modern world and its contribution to artificial intelligence. In: 2020 International conference for emerging technology (INCET). IEEE, pp 1–4
- Di Leoni PC, Lu L, Meneveau C, Karniadakis G, Zaki TA (2021) DeepONet prediction of linear instability waves in high-speed boundary layers. [arXiv:2105.08697](https://arxiv.org/abs/2105.08697)
- Dijkstra EW et al (1959) A note on two problems in connexion with graphs. *Numerische Mathematik* 1(1):269–271
- Dillon JV, Langmore I, Tran D, Brevdo E, Vasudevan S, Moore D, Patton B, Alemi A, Hoffman M, Saurous RA (2017) Tensorflow distributions. [arXiv:1711.10604](https://arxiv.org/abs/1711.10604)
- Domaneschi M, Noori AZ, Pietropinto MV, Cimellaro GP (2021) Seismic vulnerability assessment of existing school buildings. *Comput Struct* 248:106522
- Dong CZ, Catbas FN (2021) A review of computer vision-based structural health monitoring at local and global levels. *Struct Health Monit* 20(2):692–743
- du Bos ML, Balabdaoui F, Heidenreich JN (2020) Modeling stress-strain curves with neural networks: a scalable alternative to the return mapping algorithm. *Comput Mater Sci* 178:109629
- Duarte AC, Roldan F, Tubau M, Escur J, Pascual S, Salvador A, Mohedano E, McGuinness K, Torres J, Giro-i Nieto X (2019) WAV2PIX: Speech-conditioned face generation using generative adversarial networks. In: ICASSP, pp 8633–8637

- Duarte, F (2018) 5 algoritmos que ya están tomando decisiones sobre tu vida y que quizás tu no sabías [in spanish, translation: 5 algorithms that are already making decisions about your life, and perhaps you did not know]. <https://www.bbc.com/mundo/noticias-42916502>
- Duffley AH (1997) The “what” and “how” of learning in design. *IEEE Expert* 12(3):71–76
- Dumon A, Allery C, Ammar A (2011) Proper general decomposition (PGD) for the resolution of Navier-Stokes equations. *J Comput Phys* 230(4):1387–1407
- Eggersmann R, Kirchdoerfer T, Reese S, Stainier L, Ortiz M (2019) Model-free data-driven inelasticity. *Comput Methods Appl Mech Eng* 350:81–99
- Eggersmann R, Stainier L, Ortiz M, Reese S (2021) Efficient data structures for model-free data-driven computational mechanics. *Comput Methods Appl Mech Eng* 382:113855
- Eggersmann R, Stainier L, Ortiz M, Reese S (2021) Model-free data-driven computational mechanics enhanced by tensor voting. *Comput Methods Appl Mech Eng* 373:113499
- Eidnes S, Stasik AJ, Sterud C, Bøhn E, Riemer-Sørensen S (2022) Port-hamiltonian neural networks with state dependent ports. [arXiv:2206.02660](https://arxiv.org/abs/2206.02660)
- Eilers PH, Marx BD (1996) Flexible smoothing with b-splines and penalties. *Stat Sci* 11(2):89–121
- El Kadi H (2006) Modeling the mechanical behavior of fiber-reinforced polymeric composite materials using artificial neural networks—a review. *Compos Struct* 73(1):1–23
- El Said B, Hallett SR (2018) Multiscale surrogate modelling of the elastic response of thick composite structures with embedded defects and features. *Compos Struct* 200:781–798
- Erchiqui F, Kandil N (2006) Neuronal networks approach for characterization of softened polymers. *J Reinf Plast Compos* 25(5):463–473
- Erichson NB, Muehlebach M, Mahoney MW (2019) Physics-informed autoencoders for lyapunov-stable fluid flow prediction. [arXiv:1905.10866](https://arxiv.org/abs/1905.10866)
- Etedali S, Mollayi N (2018) Cuckoo search-based least squares support vector machine models for optimum tuning of tuned mass dampers. *Int J Struct Stab Dyn* 18(02):1850028
- Eubank RL (1999) Nonparametric regression and spline smoothing. CRC Press
- Farrar CR, Worden K (2012) Structural health monitoring: a machine learning perspective. Wiley, New York
- Feng N, Zhang G, Khandelwal K (2022) Finite strain FE2 analysis with data-driven homogenization using deep neural networks. *Comput Struct* 263:106742
- Fernández J, Chiachío M, Chiachío J, Muñoz R, Herrera F (2022) Uncertainty quantification in neural networks by approximate Bayesian computation: Application to fatigue in composite materials. *Eng Appl Artif Intell* 107:104511
- Fernández M, Fritzen F, Weeger O (2022) Material modeling for parametric, anisotropic finite strain hyperelasticity based on machine learning with application in optimization of metamaterials. *Int J Numer Methods Eng* 123(2):577–609. <https://onlinelibrary.wiley.com/doi/full/10.1002/nme.6869>
- Field D, Ammouche Y, Peña JM, Jérusalem A (2021) Machine learning based multiscale calibration of mesoscopic constitutive models for composite materials: application to brain white matter. *Comput Mech* 67(6):1629–1643
- Fischer CC, Tibbetts KJ, Morgan D, Ceder G (2006) Predicting crystal structure by merging data mining with quantum mechanics. *Nat Mater* 5(8):641–646
- Fish J, Wagner GJ, Keten S (2021) Mesoscopic and multiscale modelling in materials. *Nat Mater* 20(6):774–786
- Fisher RA (1936) The use of multiple measurements in taxonomic problems. *Ann Eugenics* 7(2):179–188
- Flah M, Nunez I, Ben Chaabene W, Nehdi ML (2021) Machine learning algorithms in civil structural health monitoring: a systematic review. *Arch Comput Methods Eng* 28(4):2621–2643
- Flaschel M, Kumar S, De Lorenzis L (2021) Unsupervised discovery of interpretable hyperelastic constitutive laws. *Comput Methods Appl Mech Eng* 381:113852
- Flaschel M, Kumar S, De Lorenzis L (2022) Discovering plasticity models without stress data. *NPJ Comput Mater* 8(1):1–10
- Floyd RW (1962) Algorithm 97: shortest path. *Commun ACM* 5(6):345

- Frank M, Drikakis D, Charissis V (2020) Machine-learning methods for computational science and engineering. *Computation* 8(1):15
- Frankel AL, Safta C, Alleman C, Jones R (2022) ased graph convolutional neural networks for modeling materials with microstructure. *J Mach Learn Model Comput* 3(1)
- Frankel A, Hamel CM, Bolintineanu D, Long K, Kramer S (2022) Machine learning constitutive models of elastomeric foams. *Comput Methods Appl Mech Eng* 391:114492
- Freischlad M, Schnellenbach-Held M (2005) A machine learning approach for the support of preliminary structural design. *Adv Eng Inf* 19(4):281–287
- Friedman JH (1989) Regularized discriminant analysis. *J Am Stat Assoc* 84(405):165–175
- Fu K, Li J, Zhang Y, Shen H, Tian Y (2020) Model-guided multi-path knowledge aggregation for aerial saliency prediction. *IEEE Trans Image Process* 29:7117–7127
- Fuchs A, Heider Y, Wang K, Sun W, Kaliske M (2021) DNN2: A hyper-parameter reinforcement learning game for self-design of neural network based elasto-plastic constitutive descriptions. *Comput Struct* 249:106505
- Fuhg JN, Bouklas N, Jones RE (2022) Learning hyperelastic anisotropy from data via a tensor basis neural network. [arXiv:2204.04529](https://arxiv.org/abs/2204.04529)
- Fuhg JN, Fau A, Bouklas N, Marino M (2022) Elasto-plasticity with convex model-data-driven yield functions. Hal-03619186v1. <https://hal.science/hal-03619186/>
- Fuhg JN, Böhm C, Bouklas N, Fau A, Wriggers P, Marino M (2021) Model-data-driven constitutive responses: application to a multiscale computational framework. *Int J Eng Sci* 167:103522
- Gabel J, Desaphy J, Rognan D (2014) Beware of machine learning-based scoring functions. on the danger of developing black boxes. *J Chem Inf Model* 54(10):2807–2815
- Ganin Y, Bartunov S, Li Y, Keller E, Saliceti S (2021) Computer-aided design as language. *Adv Neural Inf Process Syst* 34:5885–5897
- Gannouni S, Maad RB (2016) Numerical analysis of smoke dispersion against the wind in a tunnel fire. *J Wind Eng Ind Aerodyn* 158:61–68
- Gao K, Mei G, Piccialli F, Cuomo S, Tu J, Huo Z (2020) Julia language in machine learning: Algorithms, applications, and open issues. *Comput Sci Rev* 37:100254
- Garg A, Panigrahi BK (2021) Multi-dimensional digital twin of energy storage system for electric vehicles: A brief review. *Energy Storage* 3(6):e242
- Garg S, Gupta H, Chakraborty S (2022) Assessment of deepnet for time dependent reliability analysis of dynamical systems subjected to stochastic loading. *Eng Struct* 270:114811
- Gaurav D, Tiwari SM, Goyal A, Gandhi N, Abraham A (2020) Machine intelligence-based algorithms for spam filtering on document labeling. *Soft Comput* 24(13):9625–9638
- Gero JS (1996) Creativity, emergence and evolution in design. *Knowl-Based Syst* 9(7):435–448
- Gerolymos G, Vallet I (1996) Implicit computation of three-dimensional compressible Navier-Stokes equations using k-epsilon closure. *AIAA J* 34(7):1321–1330
- Ghosh A, SahaRay R, Chakrabarty S, Bhadra S (2021) Robust generalised quadratic discriminant analysis. *Pattern Recognit* 117:107981
- Ghoting A, Krishnamurthy R, Pednault E, Reinwald B, Sindhwani V, Tatikonda S, Tian Y, Vaithyanathan S (2011) SystemML: Declarative machine learning on mapreduce. In: 2011 IEEE 27th international conference on data engineering. IEEE, pp 231–242
- Giacinto G, Paolucci R, Roli F (1997) Application of neural networks and statistical pattern recognition algorithms to earthquake risk evaluation. *Pattern Recognit Lett* 18(11–13):1353–1362
- Gin CR, Shea DE, Brunton SL, Kutz JN (2021) DeepGreen: deep learning of Green's functions for nonlinear boundary value problems. *Sci Rep* 11(1):1–14
- Glaessgen E, Stargel D (2012) 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, p 1818
- Gobert C, Reutzel EW, Petrich J, Nassar AR, Phoha S (2018) Application of supervised machine learning for defect detection during metallic powder bed fusion additive manufacturing using high resolution imaging. *Addit Manuf* 21:517–528

- Gonzalez FJ, Balajewicz M (2018) Deep convolutional recurrent autoencoders for learning low-dimensional feature dynamics of fluid systems. [arXiv:1808.01346](https://arxiv.org/abs/1808.01346)
- González MP, Zapico JL (2008) Seismic damage identification in buildings using neural networks and modal data. *Comput Struct* 86(3–5):416–426
- González D, Chinesta F, Cueto E (2019) Learning corrections for hyperelastic models from data. *Front Mater* 6:14
- González D, Chinesta F, Cueto E (2019) Thermodynamically consistent data-driven computational mechanics. *Contin Mech Thermodyn* 31(1):239–253
- González D, García-González A, Chinesta F, Cueto E (2020) A data-driven learning method for constitutive modeling: application to vascular hyperelastic soft tissues. *Materials* 13(10):2319
- González D, Chinesta F, Cueto E (2021) Learning non-markovian physics from data. *J Comput Phys* 428:109982
- Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2020) Generative adversarial networks. *Commun ACM* 63(11):139–144
- Google cloud: AI and machine learning products. innovative machine learning products and services on a trusted platform. <https://cloud.google.com/products/ai>
- Goswami S, Anitescu C, Chakraborty S, Rabczuk T (2020) Transfer learning enhanced physics informed neural network for phase-field modeling of fracture. *Theor Appl Fract Mech* 106:102447
- Goswami S, Yin M, Yu Y, Karniadakis GE (2022) A physics-informed variational DeepONet for predicting crack path in quasi-brittle materials. *Comput Methods Appl Mech Eng* 391:114587
- Grefenstette JJ (1993) Genetic algorithms and machine learning. In: Proceedings of the sixth annual conference on computational learning theory, pp 3–4
- Greydanus S, Dzamba M, Yosinski J (2019) Hamiltonian neural networks. *Adv Neural Inf Process Syst* 32
- Gui G, Pan H, Lin Z, Li Y, Yuan Z (2017) Data-driven support vector machine with optimization techniques for structural health monitoring and damage detection. *KSCE J Civ Eng* 21(2):523–534
- Guilleminot J, Dolbow JE (2020) Data-driven enhancement of fracture paths in random composites. *Mech Res Commun* 103:103443
- Gulli A, Pal S (2017) Deep learning with Keras. Packt Publishing Ltd
- Guo J, Liu C, Cao J, Jiang D (2021) Damage identification of wind turbine blades with deep convolutional neural networks. *Renew Energy* 174:122–133
- Guo S, Agarwal M, Cooper C, Tian Q, Gao RX, Grace WG, Guo Y (2022) Machine learning for metal additive manufacturing: Towards a physics-informed data-driven paradigm. *J Manuf Syst* 62:145–163
- Hadden CM, Klimek-McDonald DR, Pineda EJ, King JA, Reichenadter AM, Miskioglu I, Gowtham S, Odegard GM (2015) Mechanical properties of graphene nanoplatelet/carbon fiber/epoxy hybrid composites: Multiscale modeling and experiments. *Carbon* 95:100–112
- Haghighat E, Juanes R (2021) Sciann: A keras/tensorflow wrapper for scientific computations and physics-informed deep learning using artificial neural networks. *Comput Methods Appl Mech Eng* 373:113552
- Haghighat E, Raissi M, Moure A, Gomez H, Juanes R (2021) A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics. *Comput Methods Appl Mech Eng* 379:113741
- Haik W, Maday Y, Chamoin L (2021) A real-time variational data assimilation method with model bias identification and correction. In: RAMSES: reduced order models; approximation theory; machine learning; surrogates; emulators and simulators
- Hall D, Llinas J (2001) Multisensor data fusion. CRC Press
- Hanifa RM, Isa K, Mohamad S (2021) A review on speaker recognition: technology and challenges. *Comput Electr Eng* 90:107005
- Hariri-Ardebili MA, Pourkamali-Anaraki F (2018) Simplified reliability analysis of multi hazard risk in gravity dams via machine learning techniques. *Arch Civ Mech Eng* 18(2):592–610

- Hasançebi O, Dumlupınar T (2013) Linear and nonlinear model updating of reinforced concrete t-beam bridges using artificial neural networks. *Comput Struct* 119:1–11
- Hashemi SM, Parvizi S, Baghbanijavid H, Tan AT, Nematollahi M, Ramazani A, Fang NX, Elahinia M (2022) Computational modelling of process-structure-property-performance relationships in metal additive manufacturing: A review. *Int Mater Rev* 67(1):1–46
- Hashemipour S, Ali M (2020) Amazon web services (AWS)—an overview of the on-demand cloud computing platform. In: *International conference for emerging technologies in computing*. Springer, Berlin, pp 40–47
- Hassan RJ, Abdulazeez AM et al (2021) Deep learning convolutional neural network for face recognition: A review. *Int J Sci Bus* 5(2):114–127
- Hastie T, Tibshirani R, Buja A (1994) Flexible discriminant analysis by optimal scoring. *J Am Stat Assoc* 89(428):1255–1270
- He Q, Laurence DW, Lee CH, Chen JS (2021) Manifold learning based data-driven modeling for soft biological tissues. *J Biomech* 117:110124
- He S, Shin HS, Tsourdos A (2021) Computational missile guidance: a deep reinforcement learning approach. *J Aerosp Inf Syst* 18(8):571–582
- He X, He Q, Chen JS (2021) Deep autoencoders for physics-constrained data-driven nonlinear materials modeling. *Comput Methods Appl Mech Eng* 385:114034
- He Q, Gu C, Valente S, Zhao E, Liu X, Yuan D (2022) Multi-arch dam safety evaluation based on statistical analysis and numerical simulation. *Sci Rep* 12(1):1–19
- Hebb DO (2005) *The organization of behavior: a neuropsychological theory*. Psychology Press
- Hemati MS, Williams MO, Rowley CW (2014) Dynamic mode decomposition for large and streaming datasets. *Phys Fluids* 26(11):111701
- Hernandez Q, Badías A, Gonzalez D, Chinesta F, Cueto E (2021) Deep learning of thermodynamics-aware reduced-order models from data. *Comput Methods Appl Mech Eng* 379:113763
- Hernández Q, Badías A, González D, Chinesta F, Cueto E (2021) Structure-preserving neural networks. *J Comput Phys* 426:109950
- Hernández Q, Badías A, Chinesta F, Cueto E (2022) Thermodynamics-informed graph neural networks. [arXiv:2203.01874](https://arxiv.org/abs/2203.01874)
- Herrada F, García-Martínez J, Fraile A, Hermanns L, Montáns F (2017) A method for performing efficient parametric dynamic analyses in large finite element models undergoing structural modifications. *Eng Struct* 131:625–638
- Hershey JR, Olsen PA (2007) Approximating the Kullback Leibler divergence between Gaussian mixture models. In: *2007 IEEE international conference on acoustics, speech and signal processing-ICASSP'07*, vol 4. IEEE, pp IV–317
- Hkdh B (1999) Neural networks in materials science. *ISIJ Int* 39(10):966–979
- Ho LV, Nguyen DH, Mousavi M, De Roeck G, Bui-Tien T, Gandomi AH, Wahab MA (2021) A hybrid computational intelligence approach for structural damage detection using marine predator algorithm and feedforward neural networks. *Comput Struct* 252:106568
- Hofmann T, Schölkopf B, Smola AJ (2008) Kernel methods in machine learning. *Ann Stat* 36(3):1171–1220
- Hong T, Wang Z, Luo X, Zhang W (2020) State-of-the-art on research and applications of machine learning in the building life cycle. *Energy Build* 212:109831
- Hong SJ, Chun H, Lee J, Kim BH, Seo MH, Kang J, Han B (2021) First-principles-based machine-learning molecular dynamics for crystalline polymers with van der waals interactions. *J Phys Chem Lett* 12(25):6000–6006
- Hoshyar AN, Samali B, Liyanapathirana R, Houshyar AN, Yu Y (2020) Structural damage detection and localization using a hybrid method and artificial intelligence techniques. *Struct Health Monit* 19(5):1507–1523
- Hosmer Jr DW, Lemeshow S, Sturdivant RX (2013) *Applied logistic regression*, vol 398. Wiley, New York
- Hou C, Wang J, Wu Y, Yi D (2009) Local linear transformation embedding. *Neurocomputing* 72(10–12):2368–2378

- Huang D, Fuhg JN, Weißenfels C, Wriggers P (2020) A machine learning based plasticity model using proper orthogonal decomposition. *Comput Methods Appl Mech Eng* 365:113008
- Ibañez R, Borzacchiello D, Aguado JV, Abisset-Chavanne E, Cueto E, Ladeveze P, Chinesta F (2017) Data-driven non-linear elasticity: constitutive manifold construction and problem discretization. *Comput Mech* 60(5):813–826
- Ibañez R, Abisset-Chavanne E, Aguado JV, Gonzalez D, Cueto E, Chinesta F (2018) A manifold learning approach to data-driven computational elasticity and inelasticity. *Arch Comput Methods Eng* 25(1):47–57
- Ibañez R, Gilormini P, Cueto E, Chinesta F (2020) Numerical experiments on unsupervised manifold learning applied to mechanical modeling of materials and structures. *Comptes Rendus. Mécanique* 348(10–11):937–958
- Ibragimova O, Brahme A, Muhammad W, Lévesque J, Inal K (2021) A new ANN based crystal plasticity model for fcc materials and its application to non-monotonic strain paths. *Int J Plast* 144:103059
- Innes M (2018) Flux: Elegant machine learning with Julia. *J Open Source Softw* 3(25):602
- Innes M, Edelman A, Fischer K, Rackauckas C, Saba E, Shah VB, Tebbutt W (2019) A differentiable programming system to bridge machine learning and scientific computing. [arXiv:1907.07587](https://arxiv.org/abs/1907.07587)
- Inoue H (2018) Data augmentation by pairing samples for images classification. [arXiv:1801.02929](https://arxiv.org/abs/1801.02929)
- Jackson NE, Webb MA, de Pablo JJ (2019) Recent advances in machine learning towards multiscale soft materials design. *Curr Opin Chem Eng* 23:106–114
- Jafari M (2020) System identification of a soil tunnel based on a hybrid artificial neural network-numerical model approach. *Iran J Sci Technol, Trans Civ Eng* 44(3):889–899
- Jagtap AD, Kharazmi E, Karniadakis GE (2020) Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. *Comput Methods Appl Mech Eng* 365:113028
- Jang DP, Fazily P, Yoon JW (2021) Machine learning-based constitutive model for J2-plasticity. *Int J Plast* 138:102919
- Jansson T, Nilsson L, Redhe M (2003) Using surrogate models and response surfaces in structural optimization—with application to crashworthiness design and sheet metal forming. *Struct Multidiscip Optim* 25(2):129–140
- Jayasundara N, Thambiratnam D, Chan T, Nguyen A (2020) Damage detection and quantification in deck type arch bridges using vibration based methods and artificial neural networks. *Eng Fail Anal* 109:104265
- Jha D, Singh S, Al-Bahrani R, Liao WK, Choudhary A, De Graef M, Agrawal A (2018) Extracting grain orientations from EBSD patterns of polycrystalline materials using convolutional neural networks. *Microsc Microanal* 24(5):497–502
- Jiang J, Hu G, Li X, Xu X, Zheng P, Stringer J (2019) Analysis and prediction of printable bridge length in fused deposition modelling based on back propagation neural network. *Virtual Phys Prototyp* 14(3):253–266
- Jiménez AA, Márquez FPG, Moraleda VB, Muñoz CQG (2019) Linear and nonlinear features and machine learning for wind turbine blade ice detection and diagnosis. *Renew Energy* 132:1034–1048
- Jiménez AA, Zhang L, Muñoz CQG, Márquez FPG (2020) Maintenance management based on machine learning and nonlinear features in wind turbines. *Renew Energy* 146:316–328
- Jin Z, Zhang Z, Demir K, Gu GX (2020) Machine learning for advanced additive manufacturing. *Matter* 3(5):1541–1556
- Jokar M, Semperlotti F (2021) Finite element network analysis: A machine learning based computational framework for the simulation of physical systems. *Comput Struct* 247:106484
- Jovanović MR, Schmid PJ, Nichols JW (2014) Sparsity-promoting dynamic mode decomposition. *Phys Fluids* 26(2):024103
- Jung J, Yoon JI, Park HK, Jo H, Kim HS (2020) Microstructure design using machine learning generated low dimensional and continuous design space. *Materialia* 11:100690

- Jung J, Yoon K, Lee PS (2020) Deep learned finite elements. *Comput Methods Appl Mech Eng* 372:113401
- Kadic M, Milton GW, van Hecke M, Wegener M (2019) 3d metamaterials. *Nat Rev Phys* 1(3):198–210
- Kaehler A, Bradski G (2016) *Learning OpenCV 3: computer vision in C++ with the OpenCV library*. O'Reilly Media, Inc
- Kalitzin G, Medic G, Iaccarino G, Durbin P (2005) Near-wall behavior of RANS turbulence models and implications for wall functions. *J Comput Phys* 204(1):265–291
- Kamath C (2001) On mining scientific datasets. In: *Data mining for scientific and engineering applications*. Springer, Berlin, pp 1–21
- Kanno Y (2018) Data-driven computing in elasticity via kernel regression. *Theor Appl Mech Lett* 8(6):361–365
- Kanouté P, Boso D, Chaboche JL, Schrefler B (2009) Multiscale methods for composites: a review. *Arch Comput Methods Eng* 16(1):31–75
- Karthikeyan J, Hie TS, Jin NY (eds) (2021) *Learning outcomes of classroom research*. L'Ordine Novo Publication, Tamil Nadu, India. ISBN: 978-93-92995-15-6
- Kao CY, Loh CH (2013) Monitoring of long-term static deformation data of fei-tsui arch dam using artificial neural network-based approaches. *Struct Control Health Monit* 20(3):282–303
- Karniadakis GE, Kevrekidis IG, Lu L, Perdikaris P, Wang S, Yang L (2021) Physics-informed machine learning. *Nat Rev Phys* 3(6):422–440
- Karpatne A, Atluri G, Faghmous JH, Steinbach M, Banerjee A, Ganguly A, Shekhar S, Samatova N, Kumar V (2017) Theory-guided data science: A new paradigm for scientific discovery from data. *IEEE Trans Knowl Data Eng* 29(10):2318–2331
- Kashinath K, Mustafa M, Albert A, Wu J, Jiang C, Esmaeilzadeh S, Azizzadenesheli K, Wang R, Chattopadhyay A, Singh A et al (2021) Physics-informed machine learning: case studies for weather and climate modelling. *Philos Trans R Soc A* 379(2194):20200093
- Khan S, Awan MJ (2018) A generative design technique for exploring shape variations. *Adv Eng Inform* 38:712–724
- Khosravi K, Shahabi H, Pham BT, Adamowski J, Shirzadi A, Pradhan B, Dou J, Ly HB, Gróf G, Ho HL et al (2019) A comparative assessment of flood susceptibility modeling using multi-criteria decision-making analysis and machine learning methods. *J Hydrol* 573:311–323
- Khurana S, Saxena S, Jain S, Dixit A (2021) Predictive modeling of engine emissions using machine learning: A review. *Mater Today: Proc* 38:280–284
- Kim P (2017) *Matlab deep learning: with machine learning, neural networks and artificial intelligence*. Apress
- Kim JW, Lee BH, Shaw MJ, Chang HL, Nelson M (2001) Application of decision-tree induction techniques to personalized advertisements on internet storefronts. *Int J Electron Commer* 5(3):45–62
- Kim C, Batra R, Chen L, Tran H, Ramprasad R (2021) Polymer design using genetic algorithm and machine learning. *Comput Mater Sci* 186:110067
- Kim Y, Park HK, Jung J, Asghari-Rad P, Lee S, Kim JY, Jung HG, Kim HS (2021) Exploration of optimal microstructure and mechanical properties in continuous microstructure space using a variational autoencoder. *Mater Des* 202:109544
- King DE (2009) Dlib-ml: a machine learning toolkit. *J Mach Learn Res* 10:1755–1758
- Kirchdoerfer T, Ortiz M (2016) Data-driven computational mechanics. *Comput Methods Appl Mech Eng* 304:81–101
- Kirchdoerfer T, Ortiz M (2017) Data driven computing with noisy material data sets. *Comput Methods Appl Mech Eng* 326:622–641
- Klein DK, Fernández M, Martin RJ, Neff P, Weeger O (2022) Polyconvex anisotropic hyperelasticity with neural networks. *J Mech Phys Solids* 159:104703
- Kleinbaum DG, Dietz K, Gail M, Klein M, Klein M (2002) *Logistic regression*. Springer, Berlin
- Ko J, Ni YQ (2005) Technology developments in structural health monitoring of large-scale bridges. *Eng Struct* 27(12):1715–1725

- Kochkov D, Smith JA, Alieva A, Wang Q, Brenner MP, Hoyer S (2021) Machine learning-accelerated computational fluid dynamics. *Proc Natl Acad Sci* 118(21):e2101784118
- Kolodziejczyk F, Mortazavi B, Rabczuk T, Zhuang X (2021) Machine learning assisted multiscale modeling of composite phase change materials for li-ion batteries' thermal management. *Int J Heat Mass Transf* 172:121199
- Kontolati K, Alix-Williams D, Boffi NM, Falk ML, Rycroft CH, Shields MD (2021) Manifold learning for coarse-graining atomistic simulations: Application to amorphous solids. *Acta Materialia* 215:117008
- Kossaifi J, Panagakis Y, Anandkumar A, Pantic M (2016) Tensorly: Tensor learning in python. [arXiv:1610.09555](https://arxiv.org/abs/1610.09555)
- Koumoulos E, Konstantopoulos G, Charitidis C (2019) Applying machine learning to nanoindentation data of (nano-) enhanced composites. *Fibers* 8(1):3
- Kralovec C, Schagerl M (2020) Review of structural health monitoring methods regarding a multi-sensor approach for damage assessment of metal and composite structures. *Sensors* 20(3):826
- Kramer MA (1991) Nonlinear principal component analysis using autoassociative neural networks. *AIChE J* 37(2):233–243
- Krzczkowski SA (1980) Measurement of liquid droplet disintegration mechanisms. *Int J Multiph Flow* 6(3):227–239
- Kulkarni KS, Kim DK, Sekar S, Samui P (2011) Model of least square support vector machine (lssvm) for prediction of fracture parameters of concrete. *Int J Concr Struct Mater* 5(1):29–33
- Ladevèze P, Néron D, Gerbaud PW (2019) Data-driven computation for history-dependent materials. *Comptes Rendus Mécanique* 347(11):831–844
- Lafamme S, Cao L, Chatzi E, Ubertini F (2016) Damage detection and localization from dense network of strain sensors. *Shock Vib* 2016:2562946
- Lakshminarayan K, Harp SA, Goldman RP, Samad T et al (1996) Imputation of missing data using machine learning techniques. In: *KDD*, vol 96. <https://cdn.aaai.org/KDD/1996/KDD96-023.pdf>
- Langley P et al (2011) The changing science of machine learning. *Mach Learn* 82(3):275–279
- Lantz B (2019) *Machine learning with R: expert techniques for predictive modeling*. Packt Publishing Ltd
- Latorre M, Montáns FJ (2013) Extension of the Sussman-Bathe spline-based hyperelastic model to incompressible transversely isotropic materials. *Comput Struct* 122:13–26
- Latorre M, Montáns FJ (2017) WYPIWYG hyperelasticity without inversion formula: Application to passive ventricular myocardium. *Comput Struct* 185:47–58
- Latorre M, Montáns FJ (2020) Experimental data reduction for hyperelasticity. *Comput Struct* 232:105919
- Latorre M, De Rosa E, Montáns FJ (2017) Understanding the need of the compression branch to characterize hyperelastic materials. *Int J Non-Linear Mech* 89:14–24
- Latorre M, Peña E, Montáns FJ (2017) Determination and finite element validation of the WYPIWYG strain energy of superficial fascia from experimental data. *Ann Biomed Eng* 45(3):799–810
- Latorre M, Mohammadkhah M, Simms CK, Montáns FJ (2018) A continuum model for tension-compression asymmetry in skeletal muscle. *J Mech Behav Biomed Mater* 77:455–460
- Le QV, Ngiam J, Coates A, Lahiri A, Prochnow B, Ng AY (2011) On optimization methods for deep learning. In: *ICML'11: Proceedings of the 28th international conference on machine learning*, pp 265–272
- Lee J, Kim J, Yun CB, Yi J, Shim J (2002) Health-monitoring method for bridges under ordinary traffic loadings. *J Sound Vib* 257(2):247–264
- Lee DW, Hong SH, Cho SS, Joo WS (2005) A study on fatigue damage modeling using neural networks. *J Mech Sci Technol* 19(7):1393–1404
- Lewandowski JJ, Seifi M (2026) Metal additive manufacturing: a review of mechanical properties. *Ann Rev of Mater Res* 46:151–186
- Lewis FL, Liu D (2013) *Reinforcement learning and approximate dynamic programming for feedback control*. Wiley, New York

- Leygue A, Coret M, Réthoré J, Stainier L, Verron E (2018) Data-based derivation of material response. *Comput Methods Appl Mech Eng* 331:184–196
- Li Z, Zhou X, Liu W, Niu Q, Kong C (2016) A similarity-based reuse system for injection mold design in automotive interior industry. *Int J Adv Manuf Technol* 87(5):1783–1795
- Li B, Huang C, Li X, Zheng S, Hong J (2019) Non-iterative structural topology optimization using deep learning. *Comput-Aided Des* 115:172–180
- Li X, Roth CC, Mohr D (2019) Machine-learning based temperature-and rate-dependent plasticity model: application to analysis of fracture experiments on dp steel. *Int J Plast* 118:320–344
- Li G, Liu Q, Zhao S, Qiao W, Ren X (2020) Automatic crack recognition for concrete bridges using a fully convolutional neural network and naive bayes data fusion based on a visual detection system. *Meas Sci Technol* 31(7):075403
- Li Y, Bao T, Chen H, Zhang K, Shu X, Chen Z, Hu Y (2021) A large-scale sensor missing data imputation framework for dams using deep learning and transfer learning strategy. *Measurement* 178:109377
- Li Y, Bao T, Chen Z, Gao Z, Shu X, Zhang K (2021) A missing sensor measurement data reconstruction framework powered by multi-task gaussian process regression for dam structural health monitoring systems. *Measurement* 186:110085
- Li Z, Kovachki N, Azizzadenesheli K, Liu B, Bhattacharya K, Stuart A, Anandkumar A (2020) Fourier neural operator for parametric partial differential equations. [arXiv:2010.08895](https://arxiv.org/abs/2010.08895)
- Lin X, Si Z, Fu W, Yang J, Guo S, Cao Y, Zhang J, Wang X, Liu P, Jiang K et al (2018) Intelligent identification of two-dimensional nanostructures by machine-learning optical microscopy. *Nano Res* 11(12):6316–6324
- Ling J, Jones R, Templeton J (2016) Machine learning strategies for systems with invariance properties. *J Comput Phys* 318:22–35
- Liu WK, Gan Z, Fleming M (2021) Knowledge-driven dimension reduction and reduced order surrogate models. In: *Mechanistic data science for stem education and applications*. Springer, Berlin, pp 131–170
- Liu J, Musialski P, Wonka P, Ye J (2012) Tensor completion for estimating missing values in visual data. *IEEE Trans Pattern Anal Mach Intell* 35(1):208–220
- Liu N, Wang Z, Sun M, Wang H, Wang B (2018) Numerical simulation of liquid droplet breakup in supersonic flows. *Acta Astronaut* 145:116–130
- Liu HH, Zhang J, Liang F, Temizel C, Basri MA, Mesdour R (2021) Incorporation of physics into machine learning for production prediction from unconventional reservoirs: A brief review of the gray-box approach. *SPE Reserv Eval Eng* 24(04):847–858
- Liu X, Tian S, Tao F, Yu W (2021) A review of artificial neural networks in the constitutive modeling of composite materials. *Compos Part B: Eng* 224:109152
- Liu B, Vu-Bac N, Zhuang X, Fu X, Rabczuk T (2022) Stochastic full-range multiscale modeling of thermal conductivity of polymeric carbon nanotubes composites: A machine learning approach. *Compos Struct* 289:115393
- Liu Y, Kutz JN, Brunton SL (2020) Hierarchical deep learning of multiscale differential equation time-steppers. [arXiv:2008.09768](https://arxiv.org/abs/2008.09768)
- Liu Y, Ponce C, Brunton SL, Kutz JN (2022) Multiresolution convolutional autoencoders. *J Comput Phys* 111801
- Liu P, Sun S (1997) The application of artificial neural networks on the health monitoring of bridges. *Structural Health Monitoring, Current Status and Perspectives*, pp 103–110
- Logarzo HJ, Capuano G, Rimoli JJ (2021) Smart constitutive laws: Inelastic homogenization through machine learning. *Comput Methods Appl Mech Eng* 373:113482
- Lopez E, Gonzalez D, Aguado J, Abisset-Chavanne E, Cueto E, Binetruy C, Chinesta F (2018) A manifold learning approach for integrated computational materials engineering. *Arch Comput Methods Eng* 25(1):59–68
- Lore KG, Stoecklein D, Davies M, Ganapathysubramanian B, Sarkar S (2015) Hierarchical feature extraction for efficient design of microfluidic flow patterns. In: *Feature extraction: modern questions and challenges*. PMLR, pp 213–225

- Lorente L, Vega J, Velazquez A (2008) Generation of aerodynamics databases using high-order singular value decomposition. *J Aircr* 45(5):1779–1788
- Lu L, Jin P, Pang G, Zhang Z, Karniadakis GE (2021) Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nat Mach Intell* 3(3):218–229
- Lu X, Liao W, Zhang Y, Huang Y (2022) Intelligent structural design of shear wall residence using physics-enhanced generative adversarial networks. *Earthq Eng Struct Dyn* 51(7):1657–1676
- Luo H, Paal SG (2019) A locally weighted machine learning model for generalized prediction of drift capacity in seismic vulnerability assessments. *Comput-Aided Civ Infrastruct Eng* 34(11):935–950
- Luo W, Hu T, Ye Y, Zhang C, Wei Y (2020) A hybrid predictive maintenance approach for CNC machine tool driven by digital twin. *Robot Comput-Integr Manuf* 65:101974
- Lynch ME, Sarkar S, Maute K (2019) Machine learning to aid tuning of numerical parameters in topology optimization. *J Mech Des* 141(11)
- Málaga-Chuquitaype C (2022) Machine learning in structural design: an opinionated review. *Frontiers Built Environ* 8:815717
- Malik M, Malik MK, Mehmood K, Makhdoom I (2021) Automatic speech recognition: a survey. *Multimed Tools Appl* 80(6):9411–9457
- Mallat S (2016) Understanding deep convolutional networks. *Philos Trans R Soc A: Math, Phys Eng Sci* 374(2065):20150203
- Manavalan M (2020) Intersection of artificial intelligence, machine learning, and internet of things-an economic overview. *Glob Discl Econ Bus* 9(2):119–128
- Mangalathu S, Jang H, Hwang SH, Jeon JS (2020) Data-driven machine-learning-based seismic failure mode identification of reinforced concrete shear walls. *Eng Struct* 208:110331
- Marr B (2019) *Artificial intelligence in practice: how 50 successful companies used AI and machine learning to solve problems*. Wiley, New York
- Martín CA, Méndez AC, Sainges O, Petiot E, Barasinski A, Piana M, Ratier L, Chinesta F (2020) Empowering design based on hybrid twin(TM): Application to acoustic resonators. *Designs* 4(4):44
- Massaroli S, Poli M, Califano F, Faragasso A, Park J, Yamashita A, Asama H (2019) Port-hamiltonian approach to neural network training. In: 2019 IEEE 58th conference on decision and control (CDC). IEEE, pp 6799–6806
- Mattheakis M, Sondak D, Dogra AS, Protopoulos P (2022) Hamiltonian neural networks for solving equations of motion. *Phys Rev E* 105(6):065305
- Maulik R, Lusch B, Balaprakash P (2021) Reduced-order modeling of advection-dominated systems with recurrent neural networks and convolutional autoencoders. *Phys Fluids* 33(3):037106
- Mayani MG, Svendsen M, Oedegaard S (2018) Drilling digital twin success stories the last 10 years. In: SPE Norway one day seminar. OnePetro
- McCoy LG, Brenna CT, Chen SS, Vold K, Das S (2022) Believing in black boxes: Machine learning for healthcare does not need explainability to be evidence-based. *J Clin Epidemiol* 142:252–257
- McCulloch WS, Pitts W (1943) A logical calculus of the ideas immanent in nervous activity. *Bull Math Biophys* 5(4):115–133
- McInnes L, Healy J, Melville J (2018) UMAP: Uniform manifold approximation and projection for dimension reduction. [arXiv:1802.03426](https://arxiv.org/abs/1802.03426)
- Mehrjoo M, Khaji N, Moharrami H, Bahreininejad A (2008) Damage detection of truss bridge joints using artificial neural networks. *Expert Syst Appl* 35(3):1122–1131
- Meijer RJ, Goeman JJ (2013) Efficient approximate k-fold and leave-one-out cross-validation for ridge regression. *Biom J* 55(2):141–155
- Meng L, Breittkopf P, Quilliec GL, Raghavan B, Villon P (2018) Nonlinear shape-manifold learning approach: concepts, tools and applications. *Arch Comput Methods Eng* 25(1):1–21
- Meng L, McWilliams B, Jarosinski W, Park HY, Jung YG, Lee J, Zhang J (2020) Machine learning in additive manufacturing: a review. *JOM* 72(6):2363–2377
- Meng X, Li Z, Zhang D, Karniadakis GE (2020) Ppinn: Parareal physics-informed neural network for time-dependent pdes. *Comput Methods Appl Mech Eng* 370:113250

- Miao P, Yokota H (2022) Comparison of markov chain and recurrent neural network in predicting bridge deterioration considering various factors. *Struct Infrastruct Eng* 1–13. <https://doi.org/10.1080/15732479.2022.2087691>
- Miao P, Yokota H, Zhang Y (2023) Deterioration prediction of existing concrete bridges using a LSTM recurrent neural network. *Struct Infrastruct* 19(4):475–489
- Michalski RS, Carbonell JG, Mitchell TM (2013) *Machine learning: An artificial intelligence approach*. Springer Science & Business Media
- Miñano M, Montáns FJ (2015) A new approach to modeling isotropic damage for Mullins effect in hyperelastic materials. *Int J Solids Struct* 67:272–282
- Miñano M, Montáns FJ (2018) WYPiWYG damage mechanics for soft materials: A data-driven approach. *Arch Comput Methods Eng* 25(1):165–193
- Mishra M (2021) Machine learning techniques for structural health monitoring of heritage buildings: A state-of-the-art review and case studies. *J Cult Herit* 47:227–245
- Mitchell M (1998) *An introduction to genetic algorithms*. MIT Press
- Miyazawa Y, Briffod F, Shiraiwa T, Enoki M (2019) Prediction of cyclic stress-strain property of steels by crystal plasticity simulations and machine learning. *Materials* 12(22):3668
- Mohammadzadeh S, Kim Y, Ahn J (2015) Pca-based neuro-fuzzy model for system identification of smart structures. *J Smart Struct Syst* 15(5):1139–1158
- Molnar C, Casalicchio G, Bischl B (2018) *iml: An R package for interpretable machine learning*. *J Open Source Softw* 3(26):786
- Monostori L, Márkus A, Van Brussel H, Westkämpfer E (1996) Machine learning approaches to manufacturing. *CIRP Ann* 45(2):675–712
- Morandin R, Nicodemus J, Unger B (2022) Port-Hamiltonian dynamic mode decomposition. [arXiv:2204.13474](https://arxiv.org/abs/2204.13474)
- Moreno S, Amores VJ, Benítez JM, Montáns FJ (2020) Reverse-engineering and modeling the 3d passive and active responses of skeletal muscle using a data-driven, non-parametric, spline-based procedure. *J Mech Behav Biomed Mater* 110:103877
- Moroni D, Pascali MA (2021) Learning topology: bridging computational topology and machine learning. *Pattern Recognit Image Anal* 31(3):443–453
- Moya B, Alfaro I, Gonzalez D, Chinesta F, Cueto E (2020) Physically sound, self-learning digital twins for sloshing fluids. *Plos One* 15(6):e0234569
- Moya B, Badías A, Alfaro I, Chinesta F, Cueto E (2022) Digital twins that learn and correct themselves. *Int J Numer Methods Eng* 123(13):3034–3044
- Mozaffar M, Bostanabad R, Chen W, Ehmann K, Cao J, Bessa M (2019) Deep learning predicts path-dependent plasticity. *Proc Natl Acad Sci* 116(52):26414–26420
- Mukherjee S, Lu D, Raghavan B, Breitkopf P, Dutta S, Xiao M, Zhang W (2021) Accelerating large-scale topology optimization: state-of-the-art and challenges. *Arch Comput Methods Eng* 28(7):4549–4571
- Muñoz D, Nadal E, Albelda J, Chinesta F, Ródenas J (2022) Allying topology and shape optimization through machine learning algorithms. *Finite Elem Anal Des* 204:103719
- Murata T, Fukami K, Fukagata K (2020) Nonlinear mode decomposition with convolutional neural networks for fluid dynamics. *J Fluid Mech* 882:(A13)1–15
- Murphy KP (2012) *Machine Learning: a probabilistic perspective*. MIT Press. ISBN 978-0262018029
- Muthali A, Laine F, Tomlin C (2021) Incorporating data uncertainty in object tracking algorithms. [arXiv:2109.10521](https://arxiv.org/abs/2109.10521)
- Nascimento RG, Viana FA (2020) Cumulative damage modeling with recurrent neural networks. *AIAA J* 58(12):5459–5471
- Nasiri S, Khosravani MR, Weinberg K (2017) Fracture mechanics and mechanical fault detection by artificial intelligence methods: A review. *Eng Fail Anal* 81:270–293
- Nassif AB, Talib MA, Nassir Q, Albadani H, Albab FD (2021) Machine learning for cloud security: a systematic review. *IEEE Access* 9:20717–20735

- Nawafleh N, AL-Oqla FM (2022) Artificial neural network for predicting the mechanical performance of additive manufacturing thermoset carbon fiber composite materials. *J Mech Behav Mater* 31(1):501–513
- Nayak HD, Anvitha L, Shetty A, D'Souza DJ, Abraham MP et al (2021) Fraud detection in online transactions using machine learning approaches—a review. *Adv Artif Intell Data Engg* 589–599
- Nayak S, Lyngdoh GA, Shukla A, Das S (2022) Predicting the near field underwater explosion response of coated composite cylinders using multiscale simulations, experiments, and machine learning. *Compos Struct* 283:115157
- Nguyen LTK, Keip MA (2018) A data-driven approach to nonlinear elasticity. *Comput Struct* 194:97–115
- Nguyen DH, Nguyen QB, Bui-Tien T, De Roeck G, Wahab MA (2020) Damage detection in girder bridges using modal curvatures gapped smoothing method and convolutional neural network: Application to bo nghi bridge. *Theor Appl Fract Mech* 109:102728
- Nguyen-Thanh VM, Zhuang X, Rabczuk T (2020) A deep energy method for finite deformation hyperelasticity. *Eur J Mech-A/Solids* 80:103874
- Ni YQ, Jiang S, Ko JM (2001) Application of adaptive probabilistic neural network to damage detection of tsing ma suspension bridge. In: *Health monitoring and management of civil infrastructure systems*, vol 4337. SPIE, pp 347–356
- Ni F, Zhang J, Noori MN (2020) Deep learning for data anomaly detection and data compression of a long-span suspension bridge. *Comput-Aided Civ Infrastruct Eng* 35(7):685–700
- Nick H, Aziminejad A, Hosseini MH, Laknejadi K (2021) Damage identification in steel girder bridges using modal strain energy-based damage index method and artificial neural network. *Eng Fail Anal* 119:105010
- Oh S, Jung Y, Kim S, Lee I, Kang N (2019) Deep generative design: Integration of topology optimization and generative models. *J Mech Des* 141(11):111405
- Olivier A, Shields MD, Graham-Brady L (2021) Bayesian neural networks for uncertainty quantification in data-driven materials modeling. *Comput Methods Appl Mech Eng* 386:114079
- Omaili A, Ismail ZH (2021) Towards machine learning for error compensation in additive manufacturing. *Appl Sci* 11(5):2375
- Ongsulee P (2017) Artificial intelligence, machine learning and deep learning. In: *2017 15th international conference on ICT and knowledge engineering (ICT&KE)*. IEEE, pp 1–6
- Paluszczek M, Thomas S (2016) *MATLAB machine learning*. Apress
- Panagiotopoulos P, Waszczyszyn Z (1999) The neural network approach in plasticity and fracture mechanics. In: *Neural networks in the analysis and design of structures*. Springer, Berlin, pp 161–195
- Panakkat A, Adeli H (2009) Recurrent neural network for approximate earthquake time and location prediction using multiple seismicity indicators. *Comput-Aided Civ Infrastruct Eng* 24(4):280–292
- Pang G, Lu L, Karniadakis GE (2019) fPINNs: Fractional physics-informed neural networks. *SIAM J Sci Comput* 41(4):A2603–A2626
- Paszczowicz W (2009) Genetic algorithms, a nature-inspired tool: survey of applications in materials science and related fields. *Mater Manuf Process* 24(2):174–197
- Pathak J, Subramanian S, Harrington P, Raja S, Chattopadhyay A, Mardani M, Kurth T, Hall D, Li Z, Azizzadenesheli K et al (2022) Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators. [arXiv:2202.11214](https://arxiv.org/abs/2202.11214)
- Pathan M, Ponnusami S, Pathan J, Pitongsawat R, Erice B, Petrinic N, Tagarielli V (2019) Predictions of the mechanical properties of unidirectional fibre composites by supervised machine learning. *Sci Rep* 9(1):1–10
- Ding S, Lin L, Wang G, Chao H, Pattern Recognit (2015) Deep feature learning with relative distance comparison for person re-identification. *Pattern Recognit* 48(10):2993–3003
- Pawar S, San O, Nair A, Rasheed A, Kvamsdal T (2021) Model fusion with physics-guided machine learning: Projection-based reduced-order modeling. *Phys Fluids* 33(6):067123

- Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V et al (2011) Scikit-learn: Machine learning in python. *J Mach Learn Res* 12:2825–2830
- Peng GC, Alber M, Buganza Tepole A, Cannon WR, De S, Dura-Bernal S, Garikipati K, Karniadakis G, Lytton WW, Perdikaris P et al (2021) Multiscale modeling meets machine learning: What can we learn? *Arch Comput Methods Eng* 28(3):1017–1037
- Penumuru DP, Muthuswamy S, Karumbu P (2020) Identification and classification of materials using machine vision and machine learning in the context of industry 4.0. *J Intell Manuf* 31(5):1229–1241
- Pereira DR, Piteri MA, Souza AN, Papa JP, Adeli H (2020) Fema: A finite element machine for fast learning. *Neural Comput Appl* 32(10):6393–6404
- Pham DT, Afify AA (2005) Machine-learning techniques and their applications in manufacturing. *Proc Inst Mech Eng, Part B: J Eng Manuf* 219(5):395–412
- Platzer A, Leygue A, Stainier L, Ortiz M (2021) Finite element solver for data-driven finite strain elasticity. *Comput Methods Appl Mech Eng* 379:113756
- Proctor JL, Brunton SL, Kutz JN (2016) Dynamic mode decomposition with control. *SIAM J Appl Dyn Syst* 15(1):142–161
- Qin J, Hu F, Liu Y, Witherell P, Wang CC, Rosen DW, Simpson T, Lu Y, Tang Q (2022) Research and application of machine learning for additive manufacturing. *Addit Manuf* 102691
- Quqa S, Martakis P, Movsessian A, Pai S, Reuland Y, Chatzi E (2022) Two-step approach for fatigue crack detection in steel bridges using convolutional neural networks. *J Civ Struct Health Monit* 12(1):127–140
- Rabin N, Fishelov D (2017) Missing data completion using diffusion maps and Laplacian pyramids. In: *International conference on computational science and its applications*. Springer, Berlin, pp 284–297
- Rai R, Sahu CK (2020) Driven by data or derived through physics? a review of hybrid physics guided machine learning techniques with cyber-physical system (CPS) focus. *IEEE Access* 8:71050–71073
- Raissi M, Karniadakis GE (2018) Hidden physics models: Machine learning of nonlinear partial differential equations. *J Comput Phys* 357:125–141
- Raissi M, Perdikaris P, Karniadakis GE (2019) Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J Comput Phys* 378:686–707
- Raj R, Tiwari MK, Ivanov D, Dolgui A (2021) Machine learning and Industry 4.0 applications. *Int J Prod Res* 59(16):4773–4778
- Ramoni M, Sebastiani P (2001) Robust learning with missing data. *Mach Learn* 45(2):147–170
- Randle D, Protopapas P, Sondak D (2020) Unsupervised learning of solutions to differential equations with generative adversarial networks. [arXiv:2007.11133](https://arxiv.org/abs/2007.11133)
- Ranković V, Grujović N, Divac D, Milivojević N, Novaković A (2012) Modelling of dam behaviour based on neuro-fuzzy identification. *Eng Struct* 35:107–113
- Rao C, Sun H, Liu Y (2021) Physics-informed deep learning for computational elastodynamics without labeled data. *J Eng Mech* 147(8):04021043
- Raschka S (2015) *Python machine learning*. Packt Publishing Ltd
- Raschka S, Mirjalili V (2019) *Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2*. Packt Publishing Ltd
- Razvi SS, Feng S, Narayanan A, Lee YTT, Witherell P (2019) A review of machine learning applications in additive manufacturing. In: *International design engineering technical conferences and computers and information in engineering conference*, vol 59179, p V001T02A040. American Society of Mechanical Engineers
- Reagan D, Sabato A, Nieszrecki C (2018) Feasibility of using digital image correlation for unmanned aerial vehicle structural health monitoring of bridges. *Struct Health Monit* 17(5):1056–1072

- Regan T, Canturk R, Slavkovsky E, Niezrecki C, Inalpolat M (2016) Wind turbine blade damage detection using various machine learning algorithms. In: International design engineering technical conferences and computers and information in engineering conference, vol 50206, p V008T10A040. American Society of Mechanical Engineers
- Regazzoni F, Dedè L, Quarteroni A (2020) Machine learning of multiscale active force generation models for the efficient simulation of cardiac electromechanics. *Comput Methods Appl Mech Eng* 370:113268
- Ren T, Wang L, Chang C, Li X (2020) Machine learning-assisted multiphysics coupling performance optimization in a photocatalytic hydrogen production system. *Energy Convers Manag* 216:112935
- Rice L, Wong E, Kolter Z (2020) Overfitting in adversarially robust deep learning. In: International conference on machine learning, pp 8093–8104. PMLR
- Rocha I, Kerfriden P, van der Meer F (2020) Micromechanics-based surrogate models for the response of composites: a critical comparison between a classical mesoscale constitutive model, hyper-reduction and neural networks. *Eur J Mech-A/Solids* 82:103995
- Rocha I, Kerfriden P, van der Meer F (2021) On-the-fly construction of surrogate constitutive models for concurrent multiscale mechanical analysis through probabilistic machine learning. *J Comput Phys: X* 9:100083
- Rodríguez M, Kramer T (2019) Machine learning of two-dimensional spectroscopic data. *Chem Phys* 520:52–60
- Rogers T, Holmes G, Cross E, Worden K (2017) On a grey box modelling framework for nonlinear system identification. In: Special topics in structural dynamics, vol 6, pp 167–178. Springer, Berlin
- Roisman I, Breitenbach J, Tropea C (2018) Thermal atomisation of a liquid drop after impact onto a hot substrate. *J Fluid Mech* 842:87–101
- Romero X, Latorre M, Montáns FJ (2017) Determination of the WYPiWYG strain energy density of skin through finite element analysis of the experiments on circular specimens. *Finite Elem Anal Des* 134:1–15
- Rosafalco L, Torzoni M, Manzoni A, Mariani S, Corigliano A (2021) Online structural health monitoring by model order reduction and deep learning algorithms. *Comput Struct* 255:106604
- Rosenblatt F (1958) The perceptron: a probabilistic model for information storage and organization in the brain. *Psychol Rev* 65(6):386
- Rosti A, Rota M, Penna A (2022) An empirical seismic vulnerability model. *Bull Earthq Eng* 20:4147–4173
- Roweis ST, Saul LK (2000) Nonlinear dimensionality reduction by locally linear embedding. *Science* 290(5500):2323–2326
- Rowley CW (2005) Model reduction for fluids, using balanced proper orthogonal decomposition. *Int J Bifurc Chaos* 15(03):997–1013
- Rubio PB, Chamoin L, Louf F (2021) Real-time data assimilation and control on mechanical systems under uncertainties. *Adv Model Simul Eng Sci* 8(1):1–25
- Rudy SH, Brunton SL, Proctor JL, Kutz JN (2017) Data-driven discovery of partial differential equations. *Sci Adv* 3(4):e1602614
- Ruggieri S, Cardellicchio A, Leggieri V, Uva G (2021) Machine-learning based vulnerability analysis of existing buildings. *Autom Constr* 132:103936
- Salazar F, Toledo M, Oñate E, Morán R (2015) An empirical comparison of machine learning techniques for dam behaviour modelling. *Struct Saf* 56:9–17
- Salazar F, Toledo MÁ, Oñate E, Suárez B (2016) Interpretation of dam deformation and leakage with boosted regression trees. *Eng Struct* 119:230–251
- Salazar F, Morán R, Toledo MÁ, Oñate E (2017) Data-based models for the prediction of dam behaviour: a review and some methodological considerations. *Arch Comput Methods Eng* 24(1):1–21

- Salloum SA, Alshurideh M, Elnagar A, Shaalan K (2020) Machine learning and deep learning techniques for cybersecurity: a review. In: The international conference on artificial intelligence and computer vision. Springer, Berlin, pp 50–57
- Salman O, Elhajj IH, Kayssi A, Chehab A (2020) A review on machine learning-based approaches for internet traffic classification. *Ann Telecommun* 75(11):673–710
- Salman O, Elhajj IH, Chehab A, Kayssi A (2022) A machine learning based framework for IoT device identification and abnormal traffic detection. *Trans Emerg Telecommun Technol* 33(3):e3743
- Sancarlos A, Cameron M, Le Peuedic JM, Groulier J, Duval JL, Cueto E, Chinesta F (2021) Learning stable reduced-order models for hybrid twins. *Data-Centric Eng* 2:e10
- Sankarasrinivasan S, Balasubramanian E, Karthik K, Chandrasekar U, Gupta R (2015) Health monitoring of civil structures with integrated uav and image processing system. *Procedia Comput Sci* 54:508–515
- Sarmadi H, Karamodin A (2020) A novel anomaly detection method based on adaptive mahalanobis-squared distance and one-class knn rule for structural health monitoring under environmental effects. *Mech Syst Signal Process* 140:106495
- Scardovelli R, Zaleski S (1999) Direct numerical simulation of free-surface and interfacial flow. *Annu Rev Fluid Mech* 31(1):567–603
- Schmid PJ (2010) Dynamic mode decomposition of numerical and experimental data. *J Fluid Mech* 656:5–28
- Schmid PJ (2011) Application of the dynamic mode decomposition to experimental data. *Exp Fluids* 50(4):1123–1130
- Schmid PJ, Li L, Juniper MP, Pust O (2011) Applications of the dynamic mode decomposition. *Theor Comput Fluid Dyn* 25(1):249–259
- Schmidt M, Lipson H (2009) Distilling free-form natural laws from experimental data. *Science* 324(5923):81–85
- Schmidt M, Lipson H (2010) Symbolic regression of implicit equations. In: Genetic programming theory and practice VII. Springer, Berlin, pp 73–85
- Schölkopf B, Herbrich R, Smola AJ (2001) A generalized representer theorem. In: International conference on computational learning theory. Springer, Berlin, pp 416–426
- Schölkopf B, Smola A, Müller KR (1997) Kernel principal component analysis. In: International conference on artificial neural networks. Springer, Berlin, pp 583–588
- Searson D (2009) GPTIPS: Genetic programming and symbolic regression for matlab. <https://sites.google.com/site/gptips4matlab/?pli=1>
- Seff A, Zhou W, Richardson N, Adams RP (2021) Vitruvion: A generative model of parametric cad sketches. [arXiv:2109.14124](https://arxiv.org/abs/2109.14124)
- Seibi A, Al-Alawi S (1997) Prediction of fracture toughness using artificial neural networks (anns). *Eng Fracture Mech* 56(3):311–319
- Sevieri G, De Falco A (2020) Dynamic structural health monitoring for concrete gravity dams based on the bayesian inference. *J Civ Struct Health Monit* 10(2):235–250
- Sharma S, Bhatt M, Sharma P (2020) Face recognition system using machine learning algorithm. In: 2020 5th international conference on communication and electronics systems (ICCES). IEEE, pp 1162–1168
- Sharp M, Ak R, Hedberg T Jr (2018) A survey of the advancing use and development of machine learning in smart manufacturing. *J Manuf Syst* 48:170–179
- Shihavuddin A, Chen X, Fedorov V, Nymark Christensen A, Andre Brogaard Riis N, Branner K, Bjorholm Dahl A, Reinhold Paulsen R (2019) Wind turbine surface damage detection by deep learning aided drone inspection analysis. *Energies* 12(4):676
- Shin D, Yoo S, Lee S, Kim M, Hwang KH, Park JH, Kang N (2021) How to trade off aesthetics and performance in generative design? In: The 2021 world congress on advances in structural engineering and mechanics (ASEM21). IASEM, KAIST, KTA, SNU DAAE

- Shorten C, Khoshgoftaar TM (2019) A survey on image data augmentation for deep learning. *J Big Data* 6(1):1–48
- Shu D, Cunningham J, Stump G, Miller SW, Yukish MA, Simpson TW, Tucker CS (2020) 3D design using generative adversarial networks and physics-based validation. *J Mech Des* 142(7):071701
- Sigmund O (2009) Systematic design of metamaterials by topology optimization. In: IUTAM symposium on modelling nanomaterials and nanosystems. Springer, Berlin, pp 151–159
- Silva M, Santos A, Figueiredo E, Santos R, Sales C, Costa JC (2016) A novel unsupervised approach based on a genetic algorithm for structural damage detection in bridges. *Eng Appl Artif Intell* 52:168–180
- Simpson T, Dervilis N, Chatzi E (2021) Machine learning approach to model order reduction of nonlinear systems via autoencoder and LSTM networks. *J Eng Mech* 147(10):04021061
- Singh AP, Medida S, Duraisamy K (2017) Machine-learning-augmented predictive modeling of turbulent separated flows over airfoils. *AIAA J* 55(7):2215–2227
- Singh H, Gupta M, Mahajan P (2017) Reduced order multiscale modeling of fiber reinforced polymer composites including plasticity and damage. *Mech Mater* 111:35–56
- Sirca G Jr, Adeli H (2012) System identification in structural engineering. *Scientia Iranica* 19(6):1355–1364
- Soize C, Ghanem R (2020) Physics-constrained non-Gaussian probabilistic learning on manifolds. *Int J Numer Methods Eng* 121(1):110–145
- Sordoni A, Bengio Y, Vahabi H, Lioma C, Grue Simonsen J, Nie JY (2015) A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In: Proceedings of the 24th ACM international conference on information and knowledge management, pp 553–562
- Sorini A, Pineda EJ, Stuckner J, Gustafson PA (2021) A convolutional neural network for multiscale modeling of composite materials. In: AIAA Scitech 2021 Forum, p 0310
- Spalart P, Allmaras S (1992) A one-equation turbulence model for aerodynamic flows. In: 30th aerospace sciences meeting and exhibit, p 439
- Speziale CG (1998) Turbulence modeling for time-dependent RANS and VLES: a review. *AIAA J* 36(2):173–184
- Sprangers O, Babuška R, Nagesh Rao SP, Lopes GA (2014) Reinforcement learning for port-Hamiltonian systems. *IEEE Trans Cybern* 45(5):1017–1027
- Stahl BC (2021) Artificial intelligence for a better future: an ecosystem perspective on the ethics of ai and emerging digital technologies. Springer Nature
- Stainier L, Leygue A, Ortiz M (2019) Model-free data-driven methods in mechanics: material data identification and solvers. *Comput Mech* 64(2):381–393
- Stančin, I., Jović, A.: An overview and comparison of free python libraries for data mining and big data analysis. In: 2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), pp. 977–982. IEEE (2019)
- Stevens B, Colonius T (2020) Enhancement of shock-capturing methods via machine learning. *Theor Comput Fluid Dyn* 34(4):483–496
- Stoll A, Benner P (2021) Machine learning for material characterization with an application for predicting mechanical properties. *GAMM-Mitteilungen* 44(1):e202100003
- Straus J, Skogestad S (2017) Variable reduction for surrogate modelling. In: Proceedings of the foundations of computer-aided process operations. Tucson, AZ, USA, pp 8–12
- Ströfer CM, Wu J, Xiao H, Paterson E (2018) Data-driven, physics-based feature extraction from fluid flow fields using convolutional neural networks. *Commun Comput Phys* 25(3):625–650
- Sun H, Burton HV, Huang H (2021) Machine learning applications for building structural design and performance assessment: State-of-the-art review. *J Build Eng* 33:101816
- Sun F, Liu Y, Sun H (2021) Physics-informed spline learning for nonlinear dynamics discovery. [arXiv:2105.02368](https://arxiv.org/abs/2105.02368)
- Surjadi JU, Gao L, Du H, Li X, Xiong X, Fang NX, Lu Y (2019) Mechanical metamaterials and their engineering applications. *Adv Eng Mater* 21(3):1800864

- Sussman T, Bathe KJ (2009) A model of incompressible isotropic hyperelastic material behavior using spline interpolations of tension-compression test data. *Commun Numer Methods Eng* 25(1):53–63
- Swischuk R, Allaire D (2019) A machine learning approach to aircraft sensor error detection and correction. *J Comput Inf Sci Eng* 19(4):041009
- Tam KMM, Moosavi V, Van Mele T, Block P (2020) Towards trans-topological design exploration of reticulated equilibrium shell structures with graph convolution networks. In: *Proceedings of IASS annual symposia, vol 2020*, pp 1–13. International Association for Shell and Spatial Structures (IASS)
- Tamke M, Nicholas P, Zwierzycki M (2018) Machine learning for architectural design: Practices and infrastructure. *Int J Arch Comput* 16(2):123–143
- Tang HS, Xue ST, Chen R, Sato T (2006) Online weighted LS-SVM for hysteretic structural system identification. *Eng Struct* 28(12):1728–1735
- Tang Q, Dang J, Cui Y, Wang X, Jia J (2022) Machine learning-based fast seismic risk assessment of building structures. *J Earthq Eng* 26(15):8041–8062
- Tartakovsky AM, Marrero CO, Perdikaris P, Tartakovsky GD, Barajas-Solano D (2018) Learning parameters and constitutive relationships with physics informed deep neural networks. [arXiv:1808.03398](https://arxiv.org/abs/1808.03398)
- Tharwat A (2016) Linear versus quadratic discriminant analysis classifier: a tutorial. *Int J Appl Pattern Recognit* 3(2):145–180
- Theocaris P, Panagiotopoulos P (1993) Neural networks for computing in fracture mechanics. methods and prospects of applications. *Comput Methods Appl Mech Eng* 106(1–2):213–228
- Ti Z, Deng XW, Yang H (2020) Wake modeling of wind turbines using machine learning. *Appl Energy* 257:114025
- Tian C, Li T, Bustillos J, Bhattacharya S, Turnham T, Yeo J, Moridi A (2021) Data-driven approaches toward smarter additive manufacturing. *Adv Intell Syst* 3(12):2100014
- Tibshirani R (1996) Regression shrinkage and selection via the LASSO. *J R Stat Soc: Ser B (Methodological)* 58(1):267–288
- Tikhonov AN (1963) On the solution of ill-posed problems and the method of regularization (in Russian). In: *Doklady Akademii Nauk*, vol 151, pp 501–504. Russian Academy of Sciences
- Torky AA, Ohno S (2021) Deep learning techniques for predicting nonlinear multi-component seismic responses of structural buildings. *Comput Struct* 252:106570
- Trinchero R, Larbi M, Torun HM, Canavero FG, Swaminathan M (2018) Machine learning and uncertainty quantification for surrogate models of integrated devices with a large number of parameters. *IEEE Access* 7:4056–4066
- Tsur EE (2020) Computer-aided design of microfluidic circuits. *Annu Rev Biomed Eng* 22:285–307
- Tu JH (2013) Dynamic mode decomposition: Theory and applications. Ph.D. thesis, Princeton University
- Turaga P, Anirudh R, Chellappa R (2020) Manifold learning. In: *Computer vision: a reference guide*. Springer, Cham. https://doi.org/10.1007/978-3-030-03243-2_824-1. https://link.springer.com/referenceworkentry/10.1007/978-3-030-03243-2_824-1
- Tzonis A, White I (2012) Automation based creative design-research and perspectives. *Newnes*
- Vafaie H, De Jong KA (1992) Genetic algorithms as a tool for feature selection in machine learning. In: *ICTAI*, pp 200–203
- Valdés-Alonzo G, Binetruy C, Eck B, García-González A, Leygue A (2022) Phase distribution and properties identification of heterogeneous materials: A data-driven approach. *Comput Methods Appl Mech Eng* 390:114354
- Van Der Schaft A, Jeltsema D, et al (2014) Port-Hamiltonian systems theory: An introductory overview. *Found Trends® Syst Control* 1(2–3):173–378
- Van Erven T, Harremoës P (2014) Rényi divergence and Kullback-Leibler divergence. *IEEE Trans Inf Theory* 60(7):3797–3820

- Vassallo D, Krishnamurthy R, Fernando HJ (2021) Utilizing physics-based input features within a machine learning model to predict wind speed forecasting error. *Wind Energy Sci* 6(1):295–309
- Verkhivker GM, Agajanian S, Hu G, Tao P (2020) Allosteric regulation at the crossroads of new technologies: multiscale modeling, networks, and machine learning. *Front Mol Biosci* 7:136
- Vitola J, Pozo F, Tibaduiza DA, Anaya M (2017) A sensor data fusion system based on k-nearest neighbor pattern classification for structural health monitoring applications. *Sensors* 17(2):417
- Vivanco-Benavides LE, Martínez-González CL, Mercado-Zúñiga C, Torres-Torres C (2022) Machine learning and materials informatics approaches in the analysis of physical properties of carbon nanotubes: a review. *Comput Mater Sci* 201:110939
- Vlassis NN, Sun W (2021) Sobolev training of thermodynamic-informed neural networks for interpretable elasto-plasticity models with level set hardening. *Comput Methods Appl Mech Eng* 377:113695
- Vlassis NN, Ma R, Sun W (2020) Geometric deep learning for computational mechanics part i: Anisotropic hyperelasticity. *Comput Methods Appl Mech Eng* 371:113299
- Volpiani PS, Meyer M, Franceschini L, Dandois J, Renac F, Martin E, Marquet O, Sipp D (2021) Machine learning-augmented turbulence modeling for RANS simulations of massively separated flows. *Phys Rev Fluids* 6(6):064607
- Wackers J, Visonneau M, Serani A, Pellegrini R, Broglia R, Diez M (2020) Multi-fidelity machine learning from adaptive-and multi-grid rans simulations. In: 33rd symposium on naval hydrodynamics
- Wang JX, Wu J, Ling J, Iaccarino G, Xiao H (2017) A comprehensive physics-informed machine learning framework for predictive turbulence modeling. [arXiv:1701.07102](https://arxiv.org/abs/1701.07102)
- Wang L et al (2020) Application and development prospect of digital twin technology in aerospace. *IFAC-PapersOnLine* 53(5):732–737
- Wang K, Sun W (2018) A multiscale multi-permeability poroplasticity model linked by recursive homogenizations and deep learning. *Comput Methods Appl Mech Eng* 334:337–380
- Wang H, O'Brien JF, Ramamoorthi R (2011) Data-driven elastic models for cloth: modeling and measurement. *ACM Trans Graph (TOG)* 30(4):1–12
- Wang L, Zhang Z, Long H, Xu J, Liu R (2016) Wind turbine gearbox failure identification with deep neural networks. *IEEE Trans Indus Inf* 13(3):1360–1368
- Wang Q, Guo Y, Yu L, Li P (2017) Earthquake prediction based on spatio-temporal data mining: an lstm network approach. *IEEE Trans Emerg Top Comput* 8(1):148–158
- Wang C, Tan X, Tor S, Lim C (2020) Machine learning in additive manufacturing: State-of-the-art and perspectives. *Addit. Manuf.* 36:101538
- Wang C, Xiao J, Zhang C, Xiao X (2020) Structural health monitoring and performance analysis of a 12-story recycled aggregate concrete structure. *Eng Struct* 205:110102
- Wang Y, Cheung SW, Chung ET, Efendiev Y, Wang M (2020) Deep multiscale model learning. *J Comput Phys* 406:109071
- Wang Y, Ghaboussi J, Hoerig C, Insana MF (2022) A data-driven approach to characterizing non-linear elastic behavior of soft materials. *J Mech Behav Biomed Mater* 130:105178
- Wang C, Xu LY, Fan JS (2020) A general deep learning framework for history-dependent response prediction based on UA-Seq2Seq model. *Comput Methods Appl Mech Eng* 372, 113357
- Waszczyszyn Z, Ziemiański L (2001) Neural networks in mechanics of structures and materials-new results and prospects of applications. *Comput Struct* 79(22–25):2261–2276
- Weiss JA, Maker BN, Govindjee S (1996) Finite element implementation of incompressible, transversely isotropic hyperelasticity. *Comput Methods Appl Mech Eng* 135(1–2):107–128
- White DA, Arrighi WJ, Kudo J, Watts SE (2019) Multiscale topology optimization using neural network surrogate models. *Comput Methods Appl Mech Eng* 346:1118–1135
- Widrow B, Hoff ME (1962) Associative storage and retrieval of digital information in networks of adaptive “neurons”. In: *Biological prototypes and synthetic systems*. Springer, Berlin, pp 160–160

- Williams MO, Kevrekidis IG, Rowley CW (2015) A data-driven approximation of the koopman operator: Extending dynamic mode decomposition. *J Nonlinear Sci* 25(6):1307–1346
- Wilt JK, Yang C, Gu GX (2020) Accelerating auxetic metamaterial design with deep learning. *Adv Eng Mater* 22(5):1901266
- Wirtz D, Karajan N, Haasdonk B (2015) Surrogate modeling of multiscale models using kernel methods. *Int J Numer Methods Eng* 101(1):1–28
- Wood MA, Cusentino MA, Wirth BD, Thompson AP (2019) Data-driven material models for atomistic simulation. *Phys Rev B* 99(18):184305
- Wu RT, Jahanshahi MR (2020) Data fusion approaches for structural health monitoring and system identification: past, present, and future. *Struct Health Monit* 19(2):552–586
- Wu Y, Sui Y, Wang G (2017) Vision-based real-time aerial object localization and tracking for uav sensing system. *IEEE Access* 5:23969–23978
- Wu JL, Xiao H, Paterson E (2018) Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework. *Phys Rev Fluids* 3(7):074602
- Wu L, Liu L, Wang Y, Zhai Z, Zhuang H, Krishnaraju D, Wang Q, Jiang H (2020) A machine learning-based method to design modular metamaterials. *Extreme Mech Lett* 36:100657
- Wu L, Zulueta K, Major Z, Arriaga A, Noels L (2020) Bayesian inference of non-linear multiscale model parameters accelerated by a deep neural network. *Comput Methods Appl Mech Eng* 360:112693
- Wu X, Park Y, Li A, Huang X, Xiao F, Usmani A (2021) Smart detection of fire source in tunnel based on the numerical database and artificial intelligence. *Fire Technol* 57(2):657–682
- Wu D, Wei Y, Terpenney J (2018) Surface roughness prediction in additive manufacturing using machine learning. In: *International manufacturing science and engineering conference*, vol 51371, p V003T02A018. American Society of Mechanical Engineers
- Xames MD, Torsha FK, Sarwar F (2023) A systematic literature review on recent trends of machine learning applications in additive manufacturing. *J Intell Manuf* 34:2529–2555
- Xiao S, Hu R, Li Z, Attarian S, Björk KM, Lendasse A (2020) A machine-learning-enhanced hierarchical multiscale method for bridging from molecular dynamics to continua. *Neural Comput Appl* 32(18):14359–14373
- Xie T, Grossman JC (2018) Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. *Phys Rev Lett* 120(14):145301
- Xie Y, Ebad Sichani M, Padgett JE, DesRoches R (2020) The promise of implementing machine learning in earthquake engineering: A state-of-the-art review. *Earthq Spectra* 36(4):1769–1801
- Xie X, Bennett J, Saha S, Lu Y, Cao J, Liu WK, Gan Z (2021) Mechanistic data-driven prediction of as-built mechanical properties in metal additive manufacturing. *NPJ Comput Mater* 7(1):1–12
- Xiong W, Wu L, Alleva F, Droppo J, Huang X, Stolcke A (2018) The Microsoft 2017 conversational speech recognition system. In: *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, pp 5934–5938
- Xu J, Duraisamy K (2020) Multi-level convolutional autoencoder networks for parametric prediction of spatio-temporal dynamics. *Comput Methods Appl Mech Eng* 372:113379
- Xu C, Cao BT, Yuan Y, Meschke G (2022) Transfer learning based physics-informed neural networks for solving inverse problems in tunneling. [arXiv:2205.07731](https://arxiv.org/abs/2205.07731)
- Xu H, Caramanis C, Mannor S (2008) Robust regression and Lasso. *Adv Neural Inf Process Syst* 21 (NIPS2008)
- Yadav D, Salmani S (2019) Deepfake: A survey on facial forgery technique using generative adversarial network. In: *2019 international conference on intelligent computing and control systems (ICCS)*. IEEE, pp 852–857
- Yagawa G, Okuda H (1996) Neural networks in computational mechanics. *Arch Comput Methods Eng* 3(4):435–512
- Yamaguchi T, Okuda H (2021) Zooming method for fea using a neural network. *Comput Struct* 247:106480

- Yan S, Zou X, Ilkhani M, Jones A (2020) An efficient multiscale surrogate modelling framework for composite materials considering progressive damage based on artificial neural networks. *Compos Part B: Eng* 194:108014
- Yan C, Vescovini R, Dozio L (2022) A framework based on physics-informed neural networks and extreme learning for the analysis of composite structures. *Comput Struct* 265:106761
- Yáñez-Márquez C (2020) Toward the bleaching of the black boxes: Minimalist machine learning. *IT Prof* 22(4):51–56
- Yang C, Kim Y, Ryu S, Gu GX (2020) Prediction of composite microstructure stress-strain curves using convolutional neural networks. *Mater Des* 189:108509
- Yang L, Zhang D, Karniadakis GE (2020) Physics-informed generative adversarial networks for stochastic differential equations. *SIAM J Sci Comput* 42(1):A292–A317
- Yang L, Meng X, Karniadakis GE (2021) B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data. *J Comput Phys* 425:109913
- Ye Y, Yang Q, Yang F, Huo Y, Meng S (2020) Digital twin for the structural health management of reusable spacecraft: A case study. *Eng Fract Mech* 234:107076
- Ye W, Hohl J, Mushongera LT (2022) Prediction of cyclic damage in metallic alloys with crystal plasticity modeling enhanced by machine learning. *Materialia* 22:101388
- Yoo S, Lee S, Kim S, Hwang KH, Park JH, Kang N (2021) Integrating deep learning into cad/cae system: generative design and evaluation of 3D conceptual wheel. *Struct Multidiscip Optim* 64:2725–2747
- Yu Y, Hur T, Jung J, Jang IG (2019) Deep learning for determining a near-optimal topological design without any iteration. *Struct Multidiscip Optim* 59(3):787–799
- Yu Y, Rashidi M, Samali B, Yousefi AM, Wang W (2021) Multi-image-feature-based hierarchical concrete crack identification framework using optimized svm multi-classifiers and d-s fusion algorithm for bridge structures. *Remote Sens* 13(2):240
- Yuan FG, Zargar SA, Chen Q, Wang S (2020) Machine learning for structural health monitoring: challenges and opportunities. *Sens Smart Struct Technol Civ, Mech, Aersp Syst* 11379:1137903
- Yuan D, Gu C, Wei B, Qin X, Xu W (2022) A high-performance displacement prediction model of concrete dams integrating signal processing and multiple machine learning techniques. *Appl Math Model* 112:436–451
- Yucel M, Bekdaş G, Nigdeli SM, Sevgen S (2019) Estimation of optimum tuned mass damper parameters via machine learning. *J Build Eng* 26:100847
- Yu S, Tack J, Mo S, Kim H, Kim J, Ha JW, Shin J (2022) Generating videos with dynamics-aware implicit generative adversarial networks. [arXiv:2202.10571](https://arxiv.org/abs/2202.10571)
- Yuvaraj P, Murthy AR, Iyer NR, Sekar S, Samui P (2013) Support vector regression based models to predict fracture characteristics of high strength and ultra high strength concrete beams. *Eng Fract Mech* 98:29–43
- Yvonnet J, He QC (2007) The reduced model multiscale method (R3M) for the non-linear homogenization of hyperelastic media at finite strains. *J Comput Phys* 223(1):341–368
- Yvonnet J, Monteiro E, He QC (2013) Computational homogenization method and reduced database model for hyperelastic heterogeneous structures. *Int J Multiscale Comput Eng* 11(3):201–225
- Zadpoor AA (2016) Mechanical meta-materials. *Mater Horiz* 3(5):371–381
- Zehtaban L, Elazhary O, Roller D (2016) A framework for similarity recognition of CAD models. *J Comput Des Eng* 3(3):274–285
- Zhan Z, Li H (2021) A novel approach based on the elastoplastic fatigue damage and machine learning models for life prediction of aerospace alloy parts fabricated by additive manufacturing. *Int J Fatigue* 145:106089
- Zhang Z, Friedrich K (2003) Artificial neural networks applied to polymer composites: a review. *Compos Sci Technol* 63(14):2029–2044

- Zhang J, Sato T, Iai S (2007) Novel support vector regression for structural system identification. *Struct Control Health Monit: Off J Int Assoc Struct Control Monit Eur Assoc Control Struct* 14(4):609–626
- Zhang Z, Hsu TY, Wei HH, Chen JH (2019) Development of a data-mining technique for regional-scale evaluation of building seismic vulnerability. *Appl Sci* 9(7):1502
- Zhang D, Guo L, Karniadakis GE (2020) Learning in modal space: Solving time-dependent stochastic PDEs using physics-informed neural networks. *SIAM J Sci Comput* 42(2):A639–A665
- Zhang XL, Michelén-Ströfer C, Xiao H (2020) Regularized ensemble Kalman methods for inverse problems. *J Comput Phys* 416:109517
- Zhang P, Yin ZY, Jin YF (2021) State-of-the-art review of machine learning applications in constitutive modeling of soils. *Arch Comput Methods Eng* 28(5):3661–3686
- Zhang Z, Liu Y (2021) Robust data-driven discovery of partial differential equations under uncertainties. [arXiv:2102.06504](https://arxiv.org/abs/2102.06504)
- Zhang W, Mehta A, Desai PS, Higgs III CF (2017) Machine learning enabled powder spreading process map for metal additive manufacturing (am). In: 2017 international solid freeform fabrication symposium. University of Texas at Austin
- Zhao Y, Akolekar HD, Weatheritt J, Michelassi V, Sandberg RD (2020) RANS turbulence model development using CFD-driven machine learning. *J Comput Phys* 411:109413
- Zhao P, Liao W, Xue H, Lu X (2022) Intelligent design method for beam and slab of shear wall structure based on deep learning. *J Build Eng* 57:104838
- Zheng H, Moosavi V, Akbarzadeh M (2020) Machine learning assisted evaluations in structural design and construction. *Autom Constr* 119:103346
- Zheng X, Zheng P, Zheng L, Zhang Y, Zhang RZ (2020) Multi-channel convolutional neural networks for materials properties prediction. *Comput Mater Sci* 173:109436
- Zheng B, Yang J, Liang B, Cheng JC (2020) Inverse design of acoustic metamaterials based on machine learning using a gauss–bayesian model. *J Appl Phys* 128(13):134902
- Zhu Y, Zabarás N, Koutsourelakis PS, Perdikaris P (2019) Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data. *J Comput Phys* 394:56–81
- Zhuang X, Guo H, Alajlan N, Zhu H, Rabczuk T (2021) Deep autoencoder based energy method for the bending, vibration, and buckling analysis of Kirchhoff plates with transfer learning. *Eur J Mech-A/Solids* 87:104225
- Zou H, Hastie T (2005) Regularization and variable selection via the elastic net. *J R Stat Soc: Ser B (Statistical Methodology)* 67(2):301–320
- zur Jacobsmühlen J, Kleszczynski S, Witt G, Merhof D (2015) Detection of elevated regions in surface images from laser beam melting processes. In: IEECON 2015-41st annual conference of the IEEE industrial electronics society. IEEE, pp 001270–001275

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

