

Machine Learning in Computer Aided Engineering

Francisco J. Montáns, Elias Cueto, Klaus-Jürgen Bathe

Abstract The extraordinary success of Machine Learning (ML) in many complex heuristic fields has promoted its introduction in more analytical engineering fields, improving or substituting many established approaches in Computer Aided Engineering (CAE), and also solving long-standing problems. In this chapter we first review the ideas behind the most used ML approaches in CAE, and then discuss a variety of different applications which have been traditionally addressed using classical approaches and that now are increasingly the focus of ML methods.

Key words: Computer Aided Engineering; Machine Learning; Classification, Identification, Prediction; Supervised, Unsupervised Learning; Training, Evaluation, Validation; Regression, Decision Trees, Support Vector Machines, Reduction, Genetic Algorithms, Neural Networks; Physics Constrained Procedures; Digital Twins; Deep Neural Operators; Constitutive, Multiscale Modeling; Surrogate Models; Finite Element Methods; Solids, Structural Mechanics, Fluids; Manufacturing, Design.

Francisco J. Montáns

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

Department of Mechanical and Aerospace Engineering, Herbert Wertheim College of Engineering, University of Florida, FL32611, USA, e-mail: fco.montans@upm.es

Elias Cueto

Aragon Institute of Engineering Research, Universidad de Zaragoza. Maria de Luna, s/n 50018-Zaragoza, Spain, e-mail: ecueto@unizar.es

Klaus-Jürgen Bathe

Mechanical Engineering Dept., Massachusetts Institute of Technology, 77 Mass. Ave., Cambridge, Mass-02139, USA, e-mail: kjb@mit.edu

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 as face recognition [211], speech [334] and speaker recognition [206], credit card fraud detection [33, 377], spam detection [14], and cloud security [375]. ML governs our specific Google searches and the advertisements we receive [271] based on our past actions, along many other interactions [1]. It even anticipates what we will type or what we will do. And, of course, ML schemes also rank us, scientists [48].

The explosion of applications of ML came with the increased computer power and the also 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 it had less presence. The success in many extremely useful areas as speech and face recognition has contributed to this interest [338]. 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 [134]. Of course, ML raises many ethical issues, some of which are described, for example in [478]. 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 [160].

Machine Learning is considered a part of Artificial Intelligence (AI) [355]. 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). The main difference with 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), 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 [411]. Nonetheless, ML codes and procedures are still mostly used as general “black-boxes”, typically employing standard implementations available in free and open-source soft-

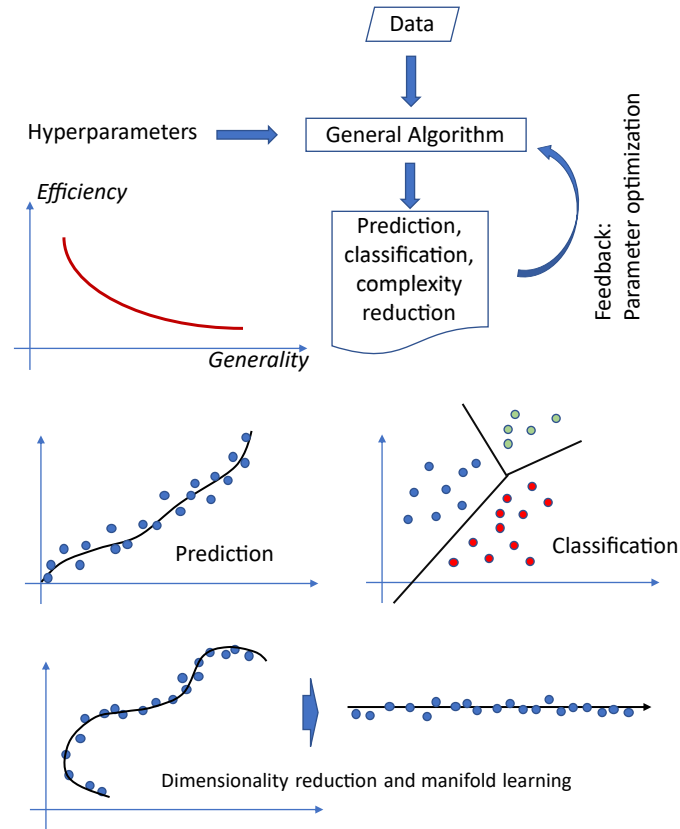


Fig. 1 Overall Machine Learning (ML) process and contrast between efficiency and generality of the method. Hyperparameters are user-defined parameters which account for the type of problem, whereas parameters are optimized for best prediction. ML may be used for prediction and for classification. It is also often used as a tool for dimensionality reduction.

ware 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 [397] (one of the best-known), Microsoft Cognitive Toolkit [550], TensorFlow [129] (which is optimal for Cuda-enabled Graphic Processing Unit (GPU) parallel ML), Keras [198], OpenNN [2], SystemML [180], just to name a few. Other proprietary software, used by big companies, are AWS Machine Learning Services from Amazon [210], Cloud Machine Learning Engine from Google [60], Matlab [389, 272], Mathematica [70, 427], etc. Moreover, many software offerings have libraries for ML, and are often used in ML projects like Python (NumPy [61], Scikit-learn [397], Tensorly [283], see review in [480]), C++, e.g., [258], Julia (a recent Just-In-Time (JIT) compiling language created with science and ML in

mind [173, 238, 239]), and the R programming environment [293, 58, 361], see also [417, 274, 173, 58]. These software offerings also use many earlier published methods for standard computational tasks 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, (usually) 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 networks concepts were rather early developed by McCulloch and Pitts in 1943 and Hebb in 1949 [217], 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. Whereas 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. Whereas 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.

2 Machine Learning procedures employed in CAE

As mentioned, ML is often considered to be a subset of AI [355, 126, 402], although often ML is also recognized as a separate field itself which only has some intersection with AI [336, 292, 388]. 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.

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.

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 [176, 113]. Examples of identification are the identification of a type of behavior or material from some stress-strain history or from force signals in machining [124, 399, 62], the identification of a nanostructure from optical microscopy [313], the identification of a person from a set of images [12, 130, 460], 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 [563, 236, 232], the prediction of a sentence from some initial words [56, 473] and the prediction of the trajectory of salient flying objects [544, 165]. 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 [408, 410]. Clustering ML procedures are typically used for classification, whereas regression ML procedures are customarily used for prediction.

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 [9]. 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

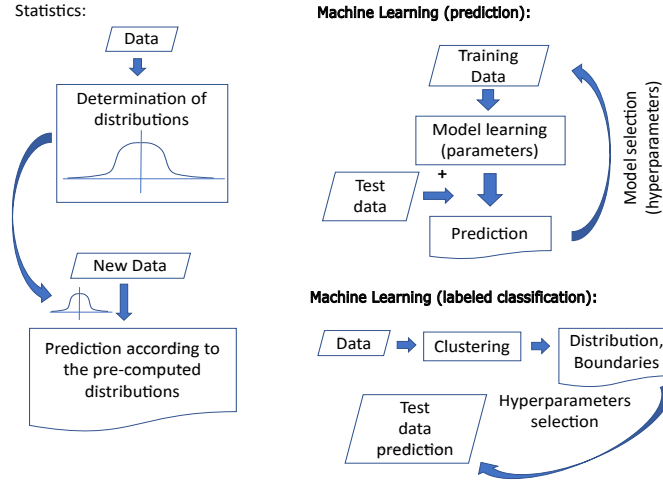


Fig. 2 Comparison of classical statistics with machine learning approaches

expect), but the purpose of DM is to research the existence of perhaps unexpected relations from raw data.

2.1.3 Statistical and optimization approaches within ML

Many procedures use, or are derived from statistics, and in particular probability theory [424, 237]. In a similar manner, ML employs mostly optimization procedures [300]. 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 as the distribution, the mean, etc. (which of course could be used thereafter for predictions), see Fig. 2. In the case of ML, the purpose is to predict new outcomes, often without the need of statistically characterizing populations, and incorporate these outcomes in further predictions [237]. 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., avoiding overfitting).

2.1.4 Supervised, unsupervised and reinforced learning

It is typical to classify the ML procedures in supervised, unsupervised, semi-supervised and reinforced learning [416, 77, 76].

In **supervised learning**, samples $\{s_i \equiv \{\mathbf{x}_i, y_i\}\}_{i=1, \dots, n} \in S$ with vectors of features \mathbf{x}_i are labeled with a known result or label y_i . The label may be a class, a 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. [433, 584, 438]. 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” [335, 182]. 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 [443, 444, 442]) 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 [291, 412, 317, 407]. 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, 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 [79, 304] with applications, for example, in the aeronautical industry [103, 489, 215].

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 data-set there may be data which are not in a logical range, or with missing entries, hence 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 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 [464]. A typical example is to perform transformations of images (rotations, translations, changes in light, etc. [240]). 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 [423]. 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.

2.1.6 Overfitting, regularization, and cross-validation

Overfitting and model complexity is an important problem in ML, see Figure 3. Given that 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. 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) [552] and L2 (called Ridge, Tikhonov-Phillips [500] or noise [59]) regularization or regression. The LASSO scheme "shrinks" the less important features (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* [593].

Model selection taking into account model fitness and including a penalization for model complexity is often performed 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 estimates the relative amount of information lost by each model, but also takes into account its parsimony. In other words, it deals with the tradeoff between overfitting and underfitting by computing

$$AIC = 2p - 2 \ln \mathfrak{L} \quad (1)$$

where p is the number of parameters of the model (complexity penalty) and \mathfrak{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).

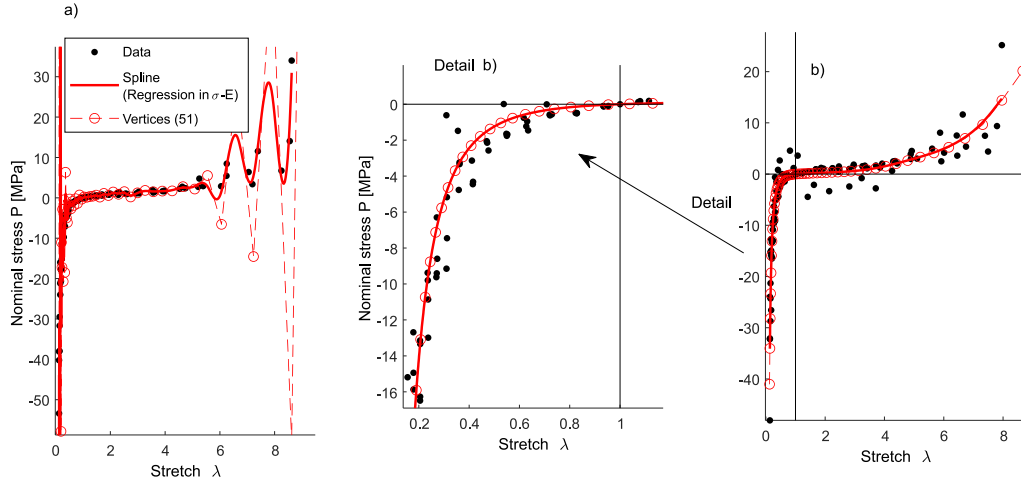


Fig. 3 Using B-splines to fit hyperelastic stress-strain data. Regression may be performed in nominal stress-stretch ($P - \lambda$) axes, or in true stress-strain $\sigma - E$ axes; note that the result is different. Whereas usual test representation in hyperelasticity is in the ($P - \lambda$) axes, 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 [298]).

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 with as few parameters as possible but also with a large probability of reproducing the data using these parameters.

Dividing the data in two sets, one for training and one for validation, very often produces overfitting, especially for small data sets. 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. 4). Leave-one-out Cross Validation (LOOCV) is the special case where the number of folds is the same as the number of samples, so the test set has only one element. Whereas LOOCV is expensive in general [348], 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 [26].

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.

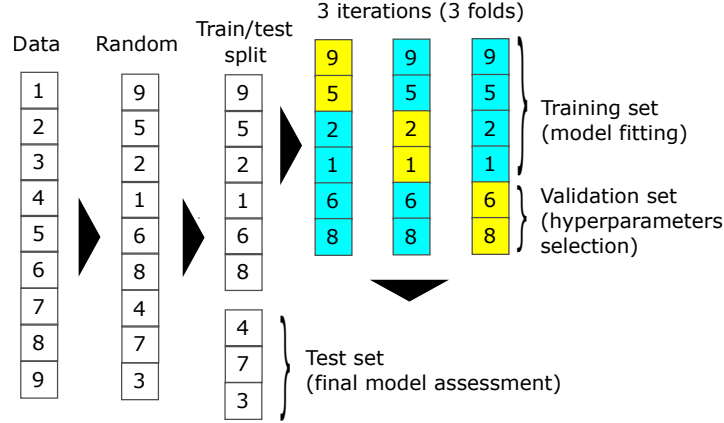


Fig. 4 Training and test sets: k-fold generation of training and validation sets from data. Number of data: 9, data for training and model selection: 6, data for final validation test (test set): 3, number of folds for model selection: 3, data in each fold: 2, number of models: 3 ($k=3$). Sometimes, the validation test is also considered as test set. The 10-fold cross validation is a common choice.

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)

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

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 [115, 297] or used to perform efficiently an inverse of functions which does not have an analytical solution [51, 147, 141]. In all these cases, smoothing techniques are fundamental to avoid overfitting.

Whereas 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{Q} . 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 $\mathfrak{L}(\mathbf{w}|(y, \mathbf{x})) = p(y|\mathbf{x}; \mathbf{w})$ 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

$$\mathfrak{L}(\mathbf{w}|(y_i, \mathbf{x}_i), i = 1, \dots, n) = \prod_{i=1}^n p(y_i | \mathbf{x}_i; \mathbf{w}) \quad (3)$$

or

$$\log \mathfrak{L}(\mathbf{w}|(y_i, \mathbf{x}_i), i = 1, \dots, n) = \sum_{i=1}^n \log p(y_i | \mathbf{x}_i; \mathbf{w}) \quad (4)$$

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

$$p(y|\mathbf{x}; \mathbf{w}, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(\mathbf{w}^T \mathbf{x} - y)^2}{2\sigma^2}\right) \quad (5)$$

it is immediate to verify that the maximization of the log-likelihood in Eq. (4) is equivalent to minimizing the MSE in Eq. (2) (regardless of the value of the variance σ^2).

A very typical regression used in ML is logistic regression [278, 230], 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

$$P(y = y_i) \equiv p^{y_i} (1 - p)^{(1-y_i)} \quad (6)$$

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

$$\text{logit}(p(\mathbf{x})) = \log\left(\frac{p(\mathbf{x})}{1 - p(\mathbf{x})}\right) = \mathbf{w}^T \mathbf{x} \quad (7)$$

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. (7) as

$$p(\mathbf{x}) = \frac{1}{1 + \exp(-\mathbf{w}^T \mathbf{x})} \quad (8)$$

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. (6) is

$$\mathcal{L}(p(\mathbf{x})|(y_i, \mathbf{x}_i), i = 1, \dots, n) = \prod_{i=1}^n p(\mathbf{x}_i)^{y_i} [1 - p(\mathbf{x}_i)]^{(1-y_i)} \quad (9)$$

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. (8) for the training data, which are a function of the parameters \mathbf{w} . The maximization of the log-likelihood of Eq. (9) for the model parameters gives the same solution as the minimization of the cross entropy

$$\arg \min_{\mathbf{w}} H(\mathbf{w}) = - \sum_{i=1}^n \left[\frac{y_i}{n} \log p(\mathbf{x}_i; \mathbf{w}) + \frac{(1-y_i)}{n} \log(1 - p(\mathbf{x}_i; \mathbf{w})) \right] \quad (10)$$

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

$$K(\mathbf{x}_i, \mathbf{x}) \equiv K_i(\mathbf{x}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left[-\frac{1}{2} \left(\frac{|\mathbf{x} - \mathbf{x}_i|}{\sigma} \right)^2 \right] \quad (11)$$

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

$$\int f^2(\mathbf{x}) d\Omega - 2 \frac{1}{n} \sum_{i=1}^n f^{(i)}(\mathbf{x}_i) \longrightarrow \min \quad (12)$$

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 (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.

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 features conditioned probabilities $p(\cdot|V)$ are obtained from the respective normal distributions. Logarithms of the probabilities are frequently used to avoid underflows.

2.2.3 Decision Trees (DT)

Decision trees are non-parametric. 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, a 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

$$G(S, x_j) = H(S) - H(S|x_j) = H(S) - \sum_{s_i = \{\mathbf{x}_i, y_i\} \in S_j} p_j H(S_j) \quad (13)$$

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

$$H(S) = - \sum_{j=1}^l p(y_j) \log p(y_j) \quad (14)$$

where $H(S)$ is the information entropy of set S for the possible labels y_j , $j = 1, \dots, l$, so Eq. (13) results in

$$G(S, x_j) = - \sum_{i=1}^l p(y_i) \log p(y_i) + \sum_{s_i \in S_j} p_j \sum_{i=1}^l p(y_i | x_j) \log p(y_i | x_j) \quad (15)$$

The gain G is computed for each feature x_j (e.g. windows, structure type, building age, 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 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 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.

Whereas 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.

2.2.4 Support Vector Machines (SVM), k-means and k-Nearest Neighbours (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

$$y \equiv f(\mathbf{x}) = \text{sign}(z(\mathbf{x})) = \text{sign}(\mathbf{w}^T \mathbf{x} + b) \quad (16)$$

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. 6. The minimization problem (in *primal form*) using Lagrange multipliers α_i is

$$\text{find } \arg \min_{\mathbf{w}, b} \left[\frac{1}{2}|\mathbf{w}|^2 + \sum_{i=1}^n \alpha_i \left(1 - y_i(\mathbf{w}^T \mathbf{x}_i + b) \right) \right] \text{ with } \alpha_i \geq 0 \quad (17)$$

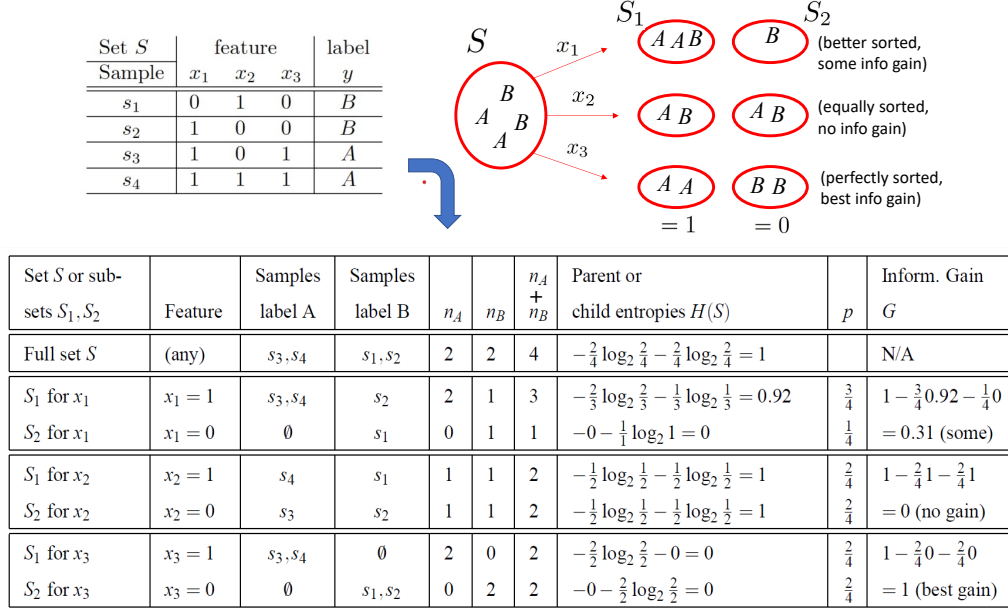


Fig. 5 Example of determination of the feature with most information gain. If we choose feature x_1 for sorting, we find two subsets, $S_1 = \{s_i \text{ such that } x_1 = 1\} = \{s_2, s_3, s_4\}$ and $S_2 = \{s_i \text{ s.t. } x_1 = 0\} = \{s_1\}$. In S_1 there are two elements (s_3, s_4) with label 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.

or in penalty form

$$\text{find } \arg \min_{\mathbf{w}, b} \left[\frac{1}{2} |\mathbf{w}|^2 + C \sum_{i=1}^n \max [0, 1 - y_i (\mathbf{w}^T \mathbf{x}_i + b)] \right] \quad (18)$$

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. 6.

Taking the derivative of the Lagrangean in squared brackets in Eq. (17) respect to \mathbf{w} and b , we get that at the minimum

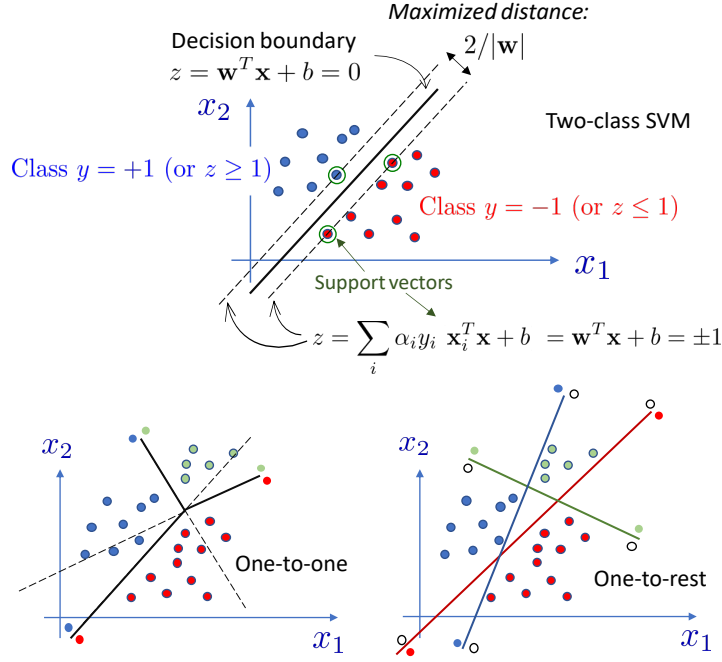


Fig. 6 Two-class SVM decision boundary and one-to-one and one-to-rest SVM multiclass classification

$$\mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i \quad \text{and} \quad \sum_{i=1}^n \alpha_i y_i = 0 \quad (19)$$

and substituting in the primal form given in Eq. (17), the minimization problem may be written in its dual form

$$\text{find } \arg \max_{\alpha_i \geq 0} \left[\sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{j,k=1}^n \alpha_j \alpha_k y_j y_k (\mathbf{x}_j^T \mathbf{x}_k) \right] \quad \text{with } \sum_{i=1}^n \alpha_i y_i = 0 \quad (20)$$

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

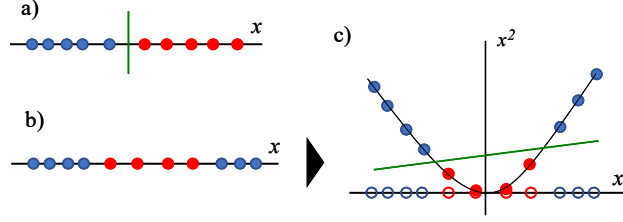


Fig. 7 Use of higher dimensions to obtain linearly separable data. a) Data is linearly separable in 1D. b) Data is not linearly separable in 1D. c) Using two dimensions with mapping $\phi = [x, x^2]^T$, data becomes linearly separable in the augmented space.

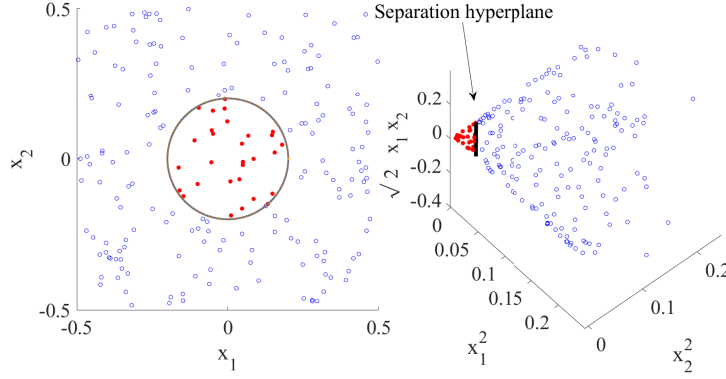


Fig. 8 Linearly non-separable samples (left). Linear separation in a transformed higher dimensional space (right)

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 separable in a larger space, see e.g. Fig. 7. This technique uses the dual form of the SVM optimization problem. Using the dual form

$$|\mathbf{w}|^2 = \mathbf{w}^T \mathbf{w} = \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j (\mathbf{x}_i^T \mathbf{x}_j) \text{ and } \mathbf{w}^T \mathbf{x} = \sum_{i=1}^n \alpha_i y_i (\mathbf{x}_i^T \mathbf{x}) \quad (21)$$

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 Figure 8 is not linearly separable-

ble 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 [454])

$$z = \sum_{i=1}^n \alpha_i y_i [\phi(\mathbf{x}_i)^T \phi(\mathbf{x})] + b \quad (22)$$

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

$$K(\mathbf{a}, \mathbf{b}) := \phi(\mathbf{a})^T \phi(\mathbf{b}) = [a_1^2 \ a_2^2 \ \sqrt{2}a_1a_2] \begin{bmatrix} b_1^2 \\ b_2^2 \\ \sqrt{2}b_1b_2 \end{bmatrix} = \left([a_1 \ a_2] \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \right)^2 = (\mathbf{a}^T \mathbf{b})^2 \quad (23)$$

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 $\phi(\mathbf{x})$, 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 $\phi(\mathbf{x})$. Then we just solve

$$\text{find } \arg \max_{\alpha_i \geq 0} \left[\sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{j,k=1}^n \alpha_j \alpha_k y_j y_k K(\mathbf{x}_j, \mathbf{x}_k) \right] \text{ with } \sum_{i=1}^n \alpha_i y_i = 0 \quad (24)$$

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)

$$K(\mathbf{a}, \mathbf{b}) = \exp(-\gamma|\mathbf{a} - \mathbf{b}|) \quad (25)$$

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 Neighbours (k-NN), a supervised technique also employed in classification. This technique uses the labels of the k -nearest neighbours to predict the label of the target points (e.g. by some weighting method).

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 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 [4, 69], Singular Value Decomposition (SVD), Proper Orthogonal Decomposition (POD) [52], Linear (Fisher's) Discriminant Analysis (LDA) [39, 155], Kernel (Nonlinear) Principal Component Analysis (kPCA), [226, 21], Local Linear Embedding (LLE) [434, 231], Manifold Learning (used also in constitutive modeling) [86, 50, 505], Uniform Manifold Approximation and Projection (UMAP) [346], and autoencoders [40, 592, 553, 75, 468]. 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

$$\text{Error}(\mathbf{w}) = \sum_{i=1}^n \left(\mathbf{x}_i - \sum_{j=1}^k w_{ij} \mathbf{x}_j \right)^2 \quad (26)$$

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

$$\mathbf{w} = \arg \min [\text{Error}(\mathbf{w})] \quad (27)$$

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

$$C(\mathbf{y}_i, i = 1, \dots, n) = \sum_{i=1}^n \left(\mathbf{y}_i - \sum_{j=1}^k w_{ij} \mathbf{y}_j \right)^2 \quad (28)$$

Isometric Mapping (ISOMAP) techniques are similar, but use geodesic k -node-to- k -node distances (computed by Dijkstra's [128] or the Floyd-Warshall [159] algorithms to find the shortest path between nodes) and look for preserving them in the reduced space. Another similar technique is the Lapacian eigenmaps scheme [49], based on the nonsingular 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 this same k-neighbours family, yet more complex and advanced, are Topological Data Analysis (TDA) techniques. A valuable overview may be found in [93]; see also the references therein.

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

$$S_{jk} = \frac{1}{n} \sum_{i=1}^n (x_{j(i)} - \bar{x}_j)(x_{k(i)} - \bar{x}_k) \quad (29)$$

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 selected as sufficient is determined by the variance ratios, see Fig. 9a. 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 [455] 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)

$$\bar{\mathbf{K}} = \mathbf{K} - \frac{1}{n} \mathbf{1} \mathbf{K} - \frac{1}{n} \mathbf{K} \mathbf{1} - \frac{1}{n^2} \mathbf{1} \mathbf{K} \mathbf{1} \quad (30)$$

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 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

$$\mathbf{S}_w = \sum_{i=1}^{n\text{-classes}} \sum_{\text{within-class-}i}^{n_i} (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^T \quad (31)$$

$$\mathbf{S}_b = \sum_{\text{Class } i=1}^{n\text{-classes}} n_i (\mathbf{m}_i - \bar{\mathbf{x}})(\mathbf{m}_i - \bar{\mathbf{x}})^T \quad (32)$$

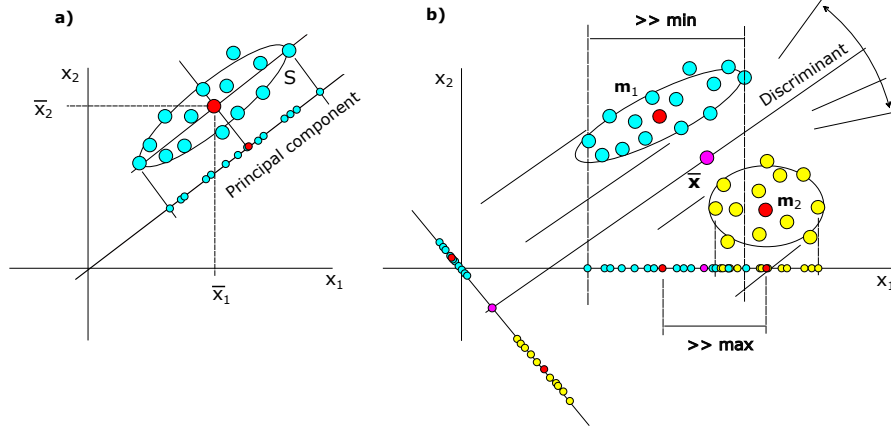


Fig. 9 a) Principal Component Analysis. The principal components are those with largest variations (largest eigenvalues of the variances matrix). b) Linear Discriminant Analysis to separate clusters. It is seen that feature x_1 is not a good choice to determine if a sample belongs to a given cluster, but there is a features combination (a line) which gives a best discrimination between clusters. That combination maximizes the distance between means of clusters whereas minimizes the dispersion of samples within the clusters

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) [495][179], Flexible Discriminant Analysis (FDA) [212] and Regularized Discriminant Analysis (RDA) [164].

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-dependnet solution as

$$\mathbf{u}(\mathbf{x}, t) = \sum_{p=1}^P a_p(t) \boldsymbol{\varphi}_p(\mathbf{x}) \quad (33)$$

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.

$$\mathbf{U} = [\mathbf{u}(\mathbf{x}, t_1) \ \mathbf{u}(\mathbf{x}, t_2) \ \dots \ \mathbf{u}(\mathbf{x}, t_n)] = \begin{bmatrix} u_{11} & \dots & u_{1n} \\ \vdots & \ddots & \vdots \\ u_{m1} & \dots & u_{mn} \end{bmatrix} \quad (34)$$

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

$$\mathbf{S} = \frac{1}{n} \mathbf{U} \mathbf{U}^T \quad (35)$$

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

$$\mathbf{u}(x_1, x_2, \dots, x_d) = \sum_{i=1}^N \boldsymbol{\phi}_i^1(x_1) \circ \boldsymbol{\phi}_i^2(x_2) \circ \dots \circ \boldsymbol{\phi}_i^d(x_d) \quad (36)$$

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

$$d_{ij}(\mathbf{x}_i, \mathbf{x}_j) = d_i^j(\mathbf{x}_i, \mathbf{x}_j) + d_j^i(\mathbf{x}_i, \mathbf{x}_j) - d_i^j(\mathbf{x}_i, \mathbf{x}_j) d_j^i(\mathbf{x}_i, \mathbf{x}_j) \quad (37)$$

where the unidirectional distance function is defined as

$$d_i^j(\mathbf{x}_i, \mathbf{x}_j) = \exp \left(- \frac{\rho_{ij} - \rho_i^1}{\rho_i^k} \right) \quad (38)$$

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 neighbour (ρ_i^1 refers to the nearest neighbour to i). Here k is an important hyperparameter. Note that $d_i^j = 1$ if i, j are nearest neighbours, and $d_i^j \rightarrow 0$ for far away neighbours. 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 [223, 510], 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)

$$KL(d(\mathbf{x}), d(\mathbf{z})) = \sum_{i,j=1}^n \left\{ d_{ij}(\mathbf{x}_i, \mathbf{x}_j) \ln \left[\frac{d_{ij}(\mathbf{x}_i, \mathbf{x}_j)}{d_{ij}(\mathbf{z}_i, \mathbf{z}_j)} \right] + [1 - d_{ij}(\mathbf{x}_i, \mathbf{x}_j)] \ln \left[\frac{1 - d_{ij}(\mathbf{x}_i, \mathbf{x}_j)}{1 - d_{ij}(\mathbf{z}_i, \mathbf{z}_j)} \right] \right\} \quad (39)$$

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

2.2.6 Genetic Algorithms

Genetic Algorithms [358] in ML (or more generally evolutionary algorithms) are in essence very similar to those employed in optimization [194, 118]. 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 [507]. The variety of implementations is large and the implementations depend on the specific problem addressed (e.g., polymer design [270], materials modeling [393], etc.), but the essence and ingredients are similar.

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. Whereas 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 [345] describing a simple model for the work of neurons, and on Rosenblatt's perceptron [432], see Fig. 10. The Adaline model [533] (Fig. 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

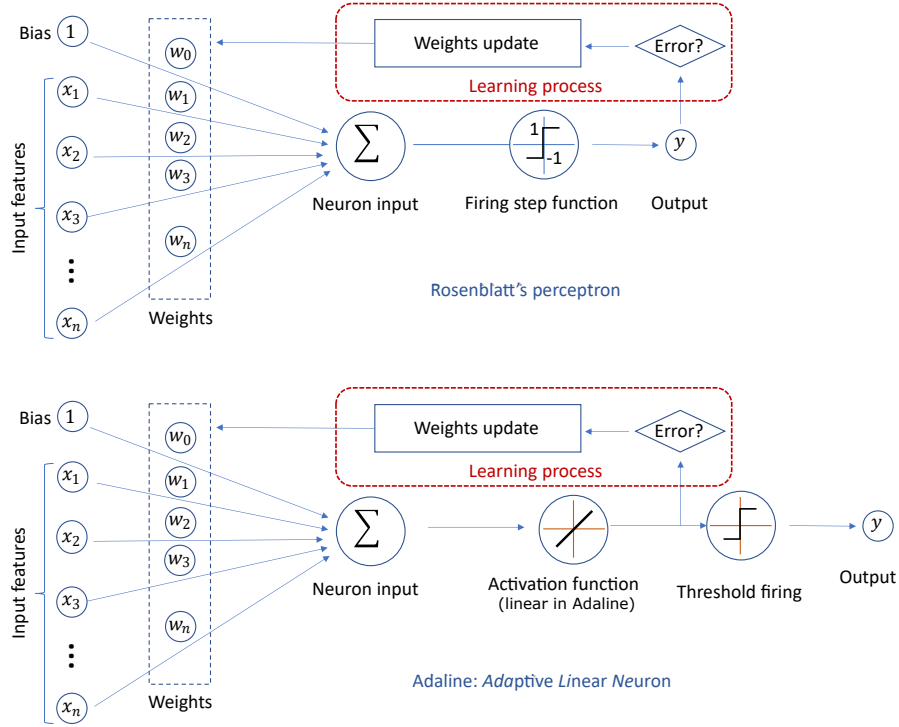


Fig. 10 Rosenblatt's perceptron and Adaline (Adaptive Linear Neuron) model.

alternatives as the ReLU (Rectified Linear Unit; the Macaulay ramp function) or the hyperbolic tangent

$$\tanh(z) = \frac{\exp(z) - \exp(-z)}{\exp(z) + \exp(-z)}, \text{ with } z = \mathbf{w}^T \mathbf{x} + w_0 \quad (40)$$

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 is defined as a composition of functions

$$\mathbf{y} = \mathbf{f}^L(\mathbf{f}^{L-1}(\mathbf{f}^{L-2}(\dots \mathbf{f}^2(\mathbf{f}^1(\mathbf{z}^1)))))) \quad (41)$$

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 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. 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. Figure 13 shows a simple case by which Rosenblath's scheme 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. 11, is trained by optimization algorithms (typically modifications of the steepest descent) using the backpropagation algorithm, which consists in computing the 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. 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

$$\Delta W_{oi}^l = -\alpha (\mathbf{y} - \mathbf{y}^{\text{exp}})^T \frac{\partial \mathbf{y}}{\partial W_{oi}^l} \quad (42)$$

where $\partial \mathbf{y} / \partial W_{oi}^l$ is computed through the chain rule. Figure 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

$$\frac{\partial y_2}{\partial W_{21}^2} = \left[\frac{\partial y_2}{\partial z_2^2} \right] \left\{ \frac{\partial z_2^2}{\partial W_{21}^2} \right\} \quad (43)$$

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. 12

$$\frac{\partial y_1}{\partial W_{21}^1} = \left[\frac{\partial y_1}{\partial z_1^2} \right] \left[\frac{z_1^2}{\partial x_2^2} \frac{\partial x_2^2}{\partial z_1^1} \right] \left\{ \frac{\partial z_2^1}{\partial W_{21}^1} \right\} \quad (44)$$

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.

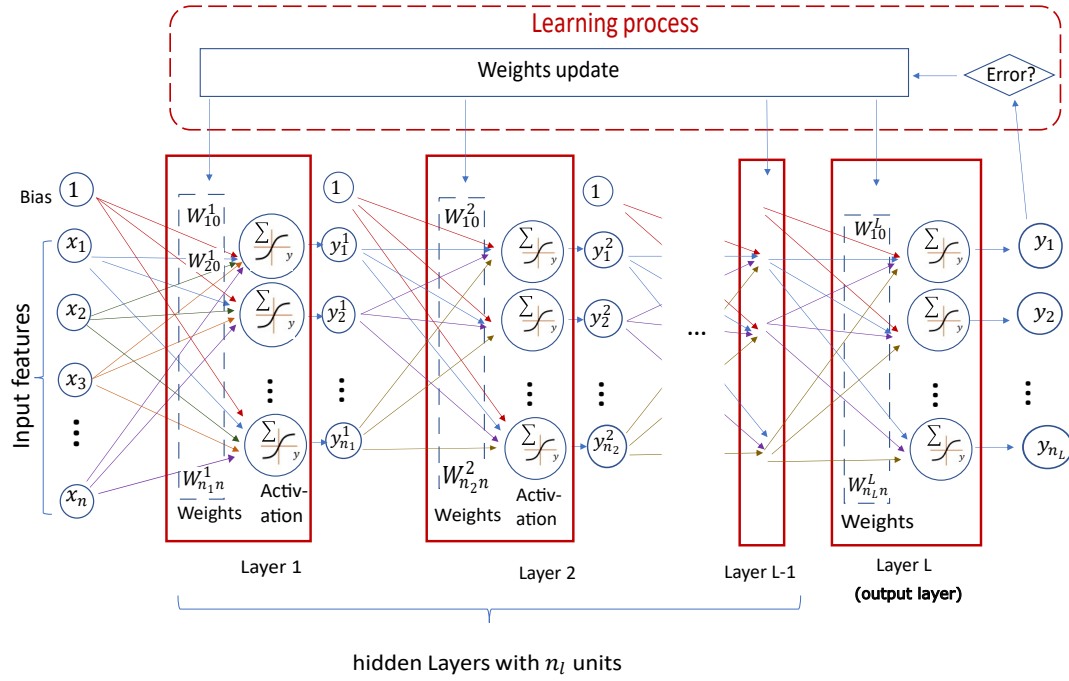


Fig. 11 Neural Network with $L-1$ hidden layers and one (the L) output layer. Notation for weights is W_{oi}^l , where i is the input cell (zero refers to the bias unit), o is the output cell (the order is often reversed in the literature), and $l = 1, \dots, L$ are the layers.

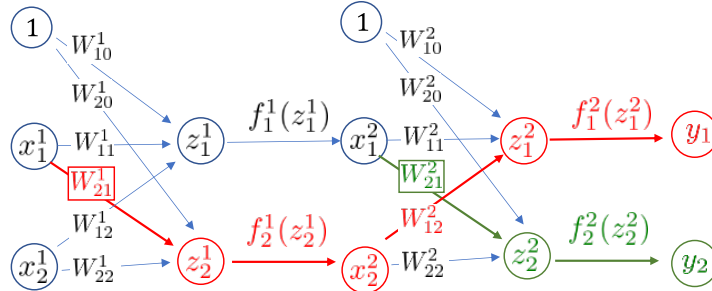


Fig. 12 Computation of the gradient through backpropagation. z_o^l is defined as $z_o^l = W_{oi}^l x_i^l$ (which includes the bias) and $f_o^l(z_o^l)$ is the activation function.

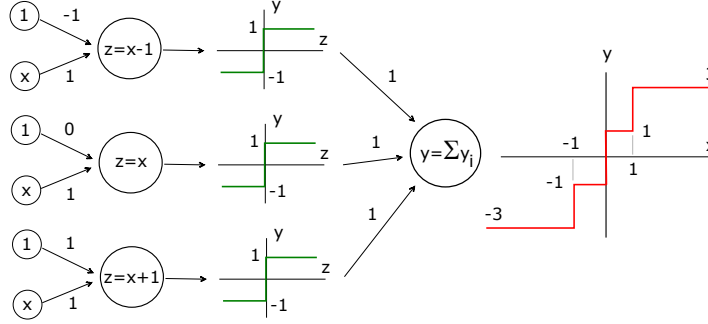


Fig. 13 Neural networks are capable of generating functions to fit data regardless of the dimension of the space and the nonlinearity of the problem. In this example, three neurons of the simplest Rosenblath's perceptron consisting of a step function, are used to generate a local linear function. This 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.

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

$$p(\mathbf{z}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{p(\mathbf{x})} \quad (45)$$

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 [386], with metal fatigue a typical application case [150, 54]. 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

$$\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^* = \arg \min_{\boldsymbol{\mu}, \boldsymbol{\Sigma}} \sum_{\forall \mathbf{x}_i, y_i} \mathcal{L}(f(\mathbf{x}_i; \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), y_i)) + \text{KL}[p(\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})), p(\mathcal{N}(0, 1))] \quad (46)$$

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

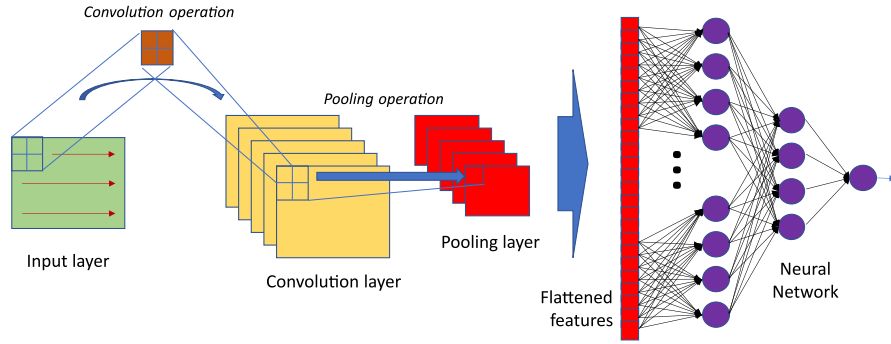


Fig. 14 Typical structure of a CNN, including one convolution layer, one pooling layer, a flattened layer of features, and a FFNN

$$y = \frac{1}{K} \sum_{k=1}^K f(\mathbf{x}; \mathcal{N}_k(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*)) \quad (47)$$

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

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. 14). Pooling is also named subsampling since performing averaging or extracting the maximum of a 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. 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. 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 [560, 162], and thus also in determining material properties [547, 590], behavior prediction [560], and obviously in extracting microstructure information from images [248].

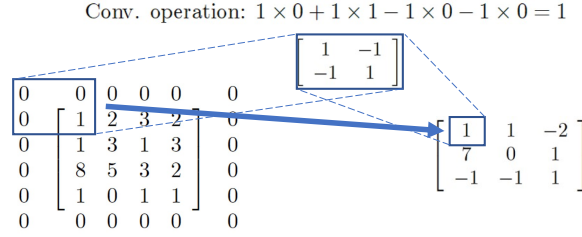


Fig. 15 Convolutional network layer with depth 1, stride length 2 (the filter patch moves 2 positions at once) and edge padding 1 (the boundary is filled with one row and column of zeroes). Pooling is similar, but usually selects the maximum or average of a moving pad to avoid correlation of features with location.

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 [368, 308, 66]). In Fig. 16 a simple RNN is shown with ${}^t\mathbf{h}$ representing the history variables, such that the equations of the RNN are

$${}^{t+1}\mathbf{h} = \mathbf{f}_h^l(\mathbf{W}_h {}^t\mathbf{h} + \mathbf{W}_x {}^t\mathbf{x} + \mathbf{b}) \quad (48)$$

$${}^t\mathbf{o} = \mathbf{f}_o^l(\mathbf{W}_h {}^t\mathbf{h} + \mathbf{W}_x {}^t\mathbf{x} + \mathbf{b}) \quad (49)$$

$${}^t\mathbf{y} = \mathbf{f}_h^l(\mathbf{W}_h {}^t\mathbf{h} + \mathbf{b}) \quad (50)$$

The unfolding of a RNN allows for a better understanding of the process, see Fig. 16. Following our previous seismic example, they can be used to study the prediction of new earthquakes from previous history, see e.g. [391, 527]. 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), many-to-many (translation into a foreign language), see Fig. 16. A frequent ingredient in RNN are “gates” (e.g., in Long Short-Term Memory (LSTM), see Fig. 17) to decide which data is introduced, output or forgotten.

2.2.11 Generative Adversarial Networks (GAN)

A Generative Adversarial Network (GAN) [191] 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

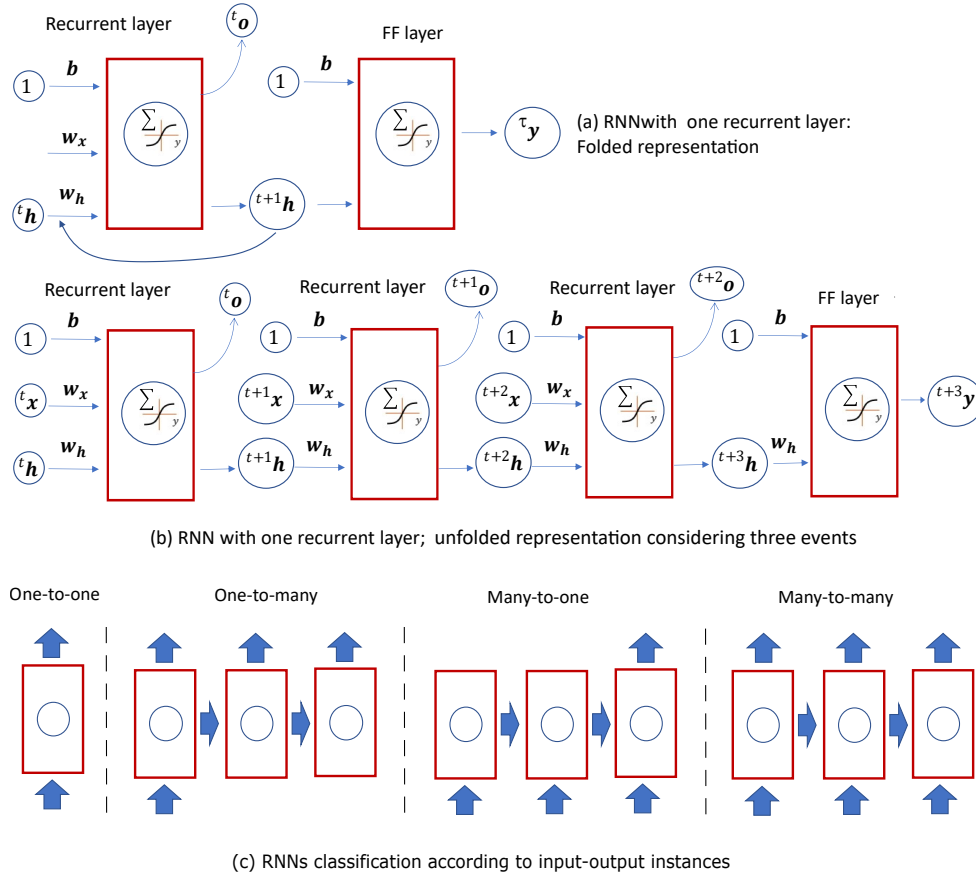


Fig. 16 Recurrent Neural Network. (a) Folded representation, (b) unfolded representation considering three events, (c) classification according to the input-output instances considered.

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 falsification of photographs (*deepfake* [554]), several images of a person are presented and the discriminator has to distinguish if they are actual pictures or manufactured photos. The technology is used to generate fake videos, and to detect them [133, 566] and is used in CAE tasks

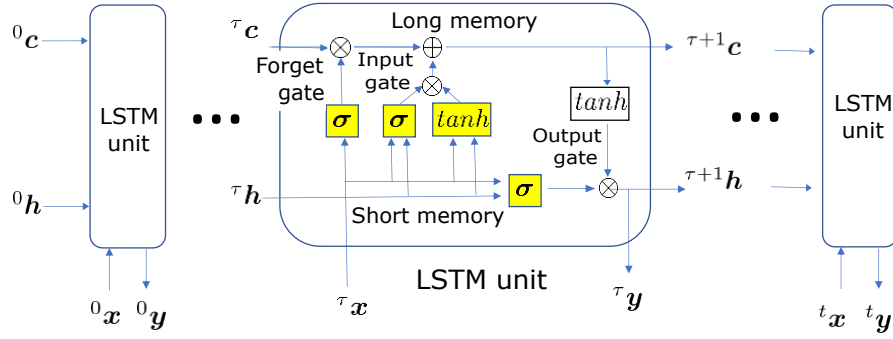


Fig. 17 A LSTM RNN, including long and short memory, and forget, input and output gates. σ is the sigmoid function, colored boxes are typical NN layers, \tanh is the hyperbolic tangent, \otimes and \oplus and \tanh are componentwise operations.

like the reconstruction of turbulent velocity fields (by comparing images) [123]. GANs are also used in the generation of compliant designs, for example in the aeronautical industry [465], and also to solve differential equations [562, 413]. A recent overview of GANs may be found in [10].

2.2.12 Ensemble Learning

Whereas 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 [181, 493]. The methods used may be, for example, shallow NNs and decision trees.

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 [344] and negative [170] attributes. Whereas 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 [31].

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) [484], or “gray-box” modeling [316, 32, 421, 428]. The simplest and probably most used method to impose such principles (an imposition called “whitening” or “bleaching” [559]) is the use of penalties and Lagrange multipliers in the cost function [122, 65, 415, 472], but there are many options and procedures to incorporate physics either in the data or in the learning [265]. 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) [264, 266], Physics Guided Machine Learning (PGML) [396, 408] 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 [264], 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, temperature) or internal (plastic or viscous deformations, damage, 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 [84, 511].

Physical principles are theoretical, of general validity, and unquestioned for the problem at hand (e.g., mass, energy, momentum conservation, 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 [37]).

3.1 Incorporating physics in, and learning physics from, the data-set

An objective may be to discover a hidden physical structure in data or physical relations in data [101]. One purpose may be to reduce the dimension of the problem by discovering relations in data that lead to the reduction of complexity [20, 17]. This is similar to calculating dynamical modes of displacements [45, 44] or to discover the invariants when relating strain components in hyperelasticity [531, 63]. Another objective may be to generate surrogate models [57, 483, 246, 320] (to discover which variables have little relevance to the physical phenomenon), or quantifying uncertainty in data [89, 502, 3, 591]. Learning physics from data is in essence a data mining approach [62, 260, 153]. 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 three-dimensional problems to plane stress or plane strain conditions. However, ML seeks an unbiased automatic approach to the solution of a problem.

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 [494]. That is the case when learning material parameters (typical in Materials Science informatics [11, 514, 482]), selecting specific functions from possibilities (e.g. selecting hardening models or hyperelastic models from a library of functions [157, 158]) 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

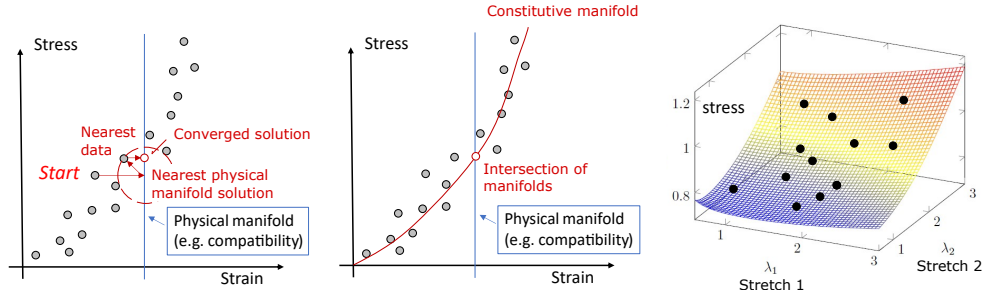


Fig. 18 Data-based constitutive modeling. Left: purely data-driven technique, where no constitutive manifold is directly employed. Instead, the closest known data point is located and is used to compute the solution. Center, right: constitutive (e.g. stress-strain) data points are used to compute a constitutive manifold (which may include uncertainty quantification), which is then employed to compute the solution in a classical manner.

internal variables as the backstress in plasticity) or the structure (as the specific existence and dependencies of variables in the yield function) [314]. Doing so, the resulting learned relations may be better interpreted and will be in compliance with our previous knowledge [6, 359, 577].

A large amount of ML methods in CAE are devoted to learning constitutive (or state) equations [305], with known conservation principles and kinematic relations (equilibrium and compatibility), as well as the boundary conditions [186, 216]. In essence, we can think of a “physical manifold” and a “constitutive manifold”, and we seek the intersection of both for some given actions or boundary and initial conditions [233, 214, 234, 380, 305]. Autoencoders are a good tool to reduce complexity and filter noise [216]. 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” [275, 276, 138, 111, 139]) 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. 18-left). Then, whereas 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 18, center and right) the manifold may be explicit (e.g. spline-based [488, 117, 297, 115, 108]) or implicit (discrete or local, e.g. [325, 235, 349, 234]). 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 [115].

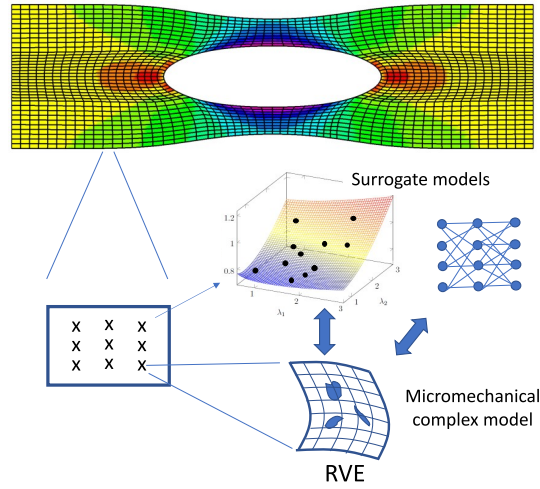


Fig. 19 Surrogate modeling representing the micromechanics. Examples are numerical manifolds or Neural Networks.

Remarkably, in some Manifold Learning approaches, physical requirements (which may include, or not, physical internal variables [23]) may result in a substantial reduction of the experimental data needed [294, 24] and of the overall computational effort, resulting also in an 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 [398, 558, 532, 143, 16, 71]. 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 [154, 30, 167]. Then, surrogate models which represent the equivalent behavior at the continuum level are extremely useful (Figure 19). These surrogate models may be obtained using different techniques. The use of Neural networks is one option [524, 528]. Then the data-set 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 [324]. Another option is to use surrogate models based on the equivalence of physical quantities as stored and dissipated energies [114, 356]. Reduced Order Methods are also important, especially in nonlinear path-dependent procedures to determine the main internal variables or simplest representation giving sufficient accuracy [470, 425]. An important aspect in surrogate modeling is the possibility of inversion of the map [203, 411, 202], 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 [273, 43, 551, 255].

3.3 Data assimilation and correction methods

The use of ML models, as when using any model (including classical analytical models, see e.g. [44]), 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) [204]. Then problems are how to incorporate new data (labeled or unlabeled) into the model [74], how to enrich the model to improve the predictions [469] and how to augment physical models with machine-learned bias [517] (hybrid models). These problems are typically encountered in dynamics [372], and the solutions are often similar to those employed in control theory [436], as the use of Kalman methods [582]. Machine learning techniques may be used for self-learning complex physical phenomena as the sloshing of fluids [366]. 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} [367]. 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

$$\mathbf{y} = f(\mathbf{x}; \mathbf{w}, \boldsymbol{\omega}) \quad (51)$$

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 [183] 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 [343] or a spacecraft simulator [564, 526] may be considered a Digital Twin) [331]. It 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 [55, 174, 78]. 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 [100] (or self-learning digital twin [366]) 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 [367] or take any other measure; that is, in essence to control a complex physical system. The dynamic equation of the Hybrid Twin is

$$\dot{\mathbf{X}}(t; \boldsymbol{\mu}) = \mathbf{A}(\mathbf{X}, t; \boldsymbol{\mu}) + \mathbf{B}(\mathbf{X}, t) + \mathbf{C}(t) + \mathbf{R}(t) \quad (52)$$

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 $\mathbf{A}(\boldsymbol{\mu})\mathbf{X}$ [445], $\mathbf{B}(\mathbf{X}, t)$ 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” [87] because analytical and data-based approaches are employed. Hybrid Twins have been applied in various fields, for example in simulating acoustic resonators [339].

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 apriori 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) [73, 437]. In this approach, the nonlinear problem

$$\dot{\mathbf{X}} = \mathbf{A}(\mathbf{X}) \quad (53)$$

is re-written as

$$\dot{\mathbf{X}} = \boldsymbol{\Theta}(\mathbf{X})\boldsymbol{\Xi} \quad (54)$$

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 [73] shown in Fig. 20, $\boldsymbol{\Theta}(\mathbf{X})$ 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 [499], although in [73] 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 [437]. Similar approaches are Physics Informed Spline Learning (PiSL) [485], which represents an improvement for data representation allowing for explicit derivatives and uses alternating direction optimization with adaptive Sequential Threshold Ridge regression (STRidge) [437] for promoting sparsity, and also more classical genetic and symbolic regression procedures [456, 453, 452]. An overview of these techniques and others may be found in [72], see also [585] for a progressive approach for considering uncertainties.

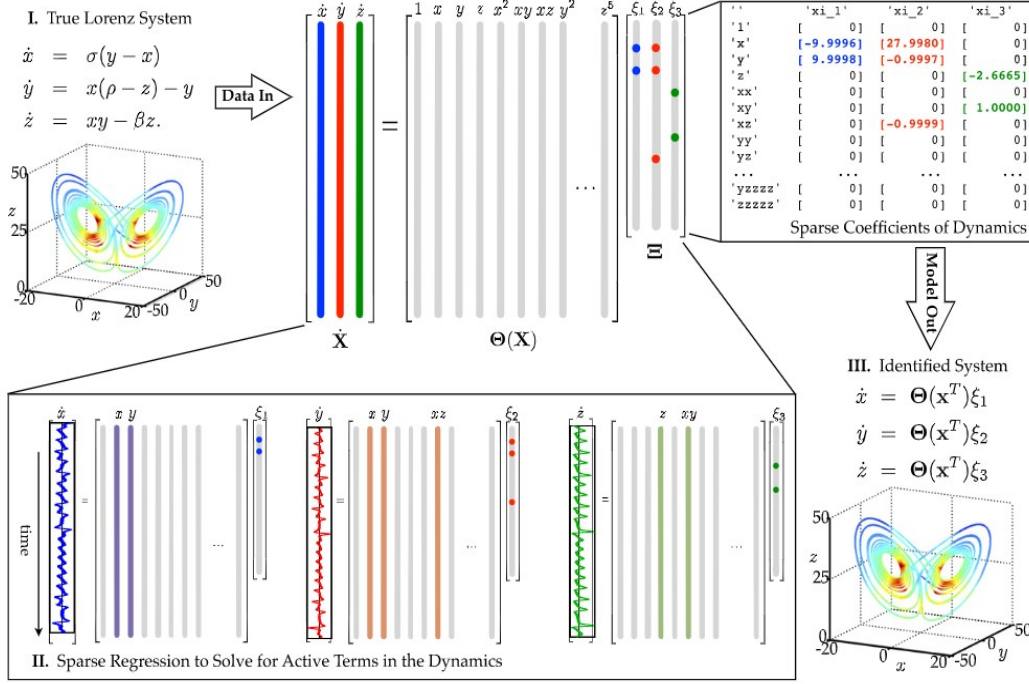


Fig. 20 Sparse Identification of Nonlinear Dynamics. Case of Lorenz System. Reproduced from [73].

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. (53)

$$\dot{\mathbf{X}} - \mathbf{A}(\mathbf{X}) = \mathbf{B}(\mathbf{X}) \quad (55)$$

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 [88]. These mixed NN approaches have also been employed in multiscale physics transferring learning through scales by increasingly deep and wide NNs [322], also employing CNNs [323]. Of course, Dynamic Mode Decomposition (DMD) [449, 504, 450, 254, 121], a procedure to determine coupled spatio-temporal modes for nonlinear problems based on Koopman (composite operator) theory [534], 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 [404].

Other techniques to discover physical relations (or nonlinear differential equations), as well as simultaneously obtain physical parameters and fields, are physics-informed NN (PINN) [410, 411, 392, 561]. 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 [351, 578]. Other applications include inverse problems in discretized conservative settings [244].

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 [96]. Based on this fact, Lu and coworkers have proposed the Deep Operator Networks (DeepONets) [328].

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 [193], instabilities in boundary layers [127], and the response of dynamical systems subjected to stochastic loadings [175].

Recently, DeepONets have been generalized by parameterizing the integral kernel in Fourier space, giving rise to the so-called Fourier Neural Operators [311]. These networks have also gained a high popularity, and have been applied to weather forecasting, for instance [394].

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

$$\dot{\mathbf{z}} = \mathbf{f}(\mathbf{z}, t), \text{ with } \mathbf{z}(0) = \mathbf{z}_0 \quad (56)$$

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

$$\dot{\mathbf{z}} = \mathbf{L}\nabla\mathbf{H} \quad (57)$$

where \mathbf{L} is the classical (skew-symmetric) symplectic matrix, ensures that the learnt dynamics will conserve energy, because it is derived from the Hamiltonian \mathbf{H} . Many recent references have exploited this approach, either in Hamiltonian or Lagrangian frameworks [195, 341, 112]. 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

$$\dot{\mathbf{z}} = \mathbf{L}\nabla\mathbf{H} + \mathbf{M}\nabla\mathbf{S} \quad (58)$$

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

$$\mathbf{L}\nabla\mathbf{S} = \mathbf{M}\nabla\mathbf{H} = \mathbf{0} \quad (59)$$

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 [221, 219].

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.

4.1 Constitutive modeling and multiscale applications

A 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” [556]. There are several reviews of applications of ML in constitutive modeling (specially using NNs), in continuum mechanics [62], for soils [580], composites [583, 321, 142], and material science [224]. An earlier review of NN applications in computational mechanics in general can also be found in [555]. Below we briefly review some applications.

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 [275, 110], and even earlier in [522] for cloths in the animation and design industries. Data-driven nonlinear elasticity is also analyzed in several works [111, 479, 380], and applied to soft tissues [188] and foams [161].

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 [138, 139, 403]. Kernel regression is also employed [261].

Another common methodology is the use of data-driven constitutive manifolds [234], where identification and reduction of the constitutive manifolds allow for a much more efficient approach. NNs are as well used in finite deformation elasticity [381, 529].

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 [186, 101, 220]. 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 [157]). Then, this energy may be modelled, for example, by splines or B-splines. This approach, based on the Valanis-Landel assumption, was pioneered by Sussman and Bathe for isotropic polymers [488] and extended later for anisotropic materials [296] like soft biological tissues (fascia, [299], skin [430], heart [297], muscle [295, 364]), compressible materials [115], auxetic foams [116, 114] and composites [25]. Polynomials in terms of invariants are also employed, with the coefficients determined by sparse regression [157]. Another approach is to select models from a database, and possibly correct them [185, 144], or select specific function models for the hyperelastic stored energy using machine learning methods (e.g. NNs) [157, 515, 381]. In particular, polyconvexity (to guarantee stability and

global minimizers for the elastic boundary-value problem) may also be imposed in NN models [277]. Anisotropy in hyperelasticity may be learned from data with NNs [168].

In material data-sets, 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 distance to the solution and using an estimation based on maximum entropy [276]. For spline-based constitutive modeling, experimental data reduction using stability-based penalizations allows for the use of noisy datasets and outliers avoiding overfitting [298].

4.1.2 Plasticity, viscoelasticity, 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 [187]. However, some early works using NNs are available [390]. 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 [352] and anisotropic materials [356]. Crack growth in the aircraft industry has also been determined with RNNs [373]. Of course, ML has been for a long time applied to model fatigue [301].

Plasticity is probably the most studied case of the non-conservative behaviors [530]. 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 [289], see also [233].

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) [137, 106]. 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 [245]. In general, RNNs [368, 65] and CNNs [6] 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 [521].

In materials science, ML is employed to predict the cyclic stress-strain behavior depending on the microstructure of the material obtained from EBSD (electron backscatter diffraction) 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) [158]. A mixture of analytical formulas and FFNNs machine learning has been employed to replace the temperature- and rate-dependent term of the Johnson-Cook model [308]. 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 [516]. 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 [169]. Automatic hyperparameter (self-)learning has been addressed for NN modeling of elastoplasticity in [166].

4.1.3 Fracture

Fracture phenomena may also be modelled using NNs [496, 458] and data-driven model-free techniques [85]. Data-driven model extraction from experimental data and knowledge transfer [192] have been applied to obtain predictions in 3D models from 2D cases [320]. Data-driven approaches are used to enhance fracture paths in simulations of random composites from isotropic cases a model reduction to avoid high fidelity phase-field computations [197]. SVMs and variants have been used for predicting fracture properties, e.g. [572, 288], and so have been other methods like BNN, Genetic Algorithm (GA) and hybrid systems, see e.g. [374, 229]

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 [398], in particular when nonlinear behavior is modelled [241]. This is particularly the case when the macroscopic behavior or the physical properties depend in a complex manner on a specific microstructure [154] or on physics equations and phenomena only seen at a micro- or smaller scale, as atomistic [80, 282, 537], molecular [546] or cellular [512].

ML allows for the simpler implementation of first-principles in multiscale simulations [227], 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) [92]. Generating surrogate models to reproduce the observed macroscopic effects due to complex phenomena at the microscale [536] is often only possible through ML and Model Order Reduction (MOR) [528, 573]. Even in the simplest cases, ML may substantially speed-up the expensive computational costs of classical nonlinear FE2 homogenization techniques [149, 541], allowing for real-time simulations [426]. 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) [574]. Material designs are often obtained from inverse analyses facilitated by parametric ML surrogate models [241, 203]. In particular, ML may be employed to determine the phase distributions in heterogeneous materials [508].

The modeling of classical fiber-based and complex composite heterogeneous materials often requires multiscale approaches [395, 201, 262] 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 [315]. Modeling of complex properties such as composite phase changes for thermal management in Li-ion batteries may be performed with CNNs [281]. Indeed, CNNs can also be used for performing an inverse analysis [474]. 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. [152, 378, 284].

4.1.5 Metamaterials modeling

Metamaterials are architected materials with inner custom-made structure. With the current development of 3D printing, metamaterials modeling and design is becoming an important field [257, 53, 575, 42] because a material with unique salient properties may be designed ad libitum allowing for a wide range of applications [487]. Their design has evolved from the classical optimization-based approach [466]. 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. [540, 151, 588, 535].

4.2 Fluid mechanics applications

Fluid phenomena and related modeling approaches are very rich, spanning from the breakup of liquid droplets under different conditions [287, 429, 318] to smoke from fires in tunnels [172, 543], emissions from engines [269, 38], flow and wake effects in wind turbines [107, 497], and free surface flow dynamics [47, 448]. 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.

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 $\rho \bar{\tilde{\mathbf{u}} \otimes \tilde{\mathbf{u}}}$

$$\nabla \bar{\mathbf{u}} \cdot \bar{\mathbf{u}} = \frac{1}{\rho} \nabla (-p \mathbf{I} + 2\mu \nabla^s \bar{\mathbf{u}} - \rho \bar{\tilde{\mathbf{u}} \otimes \tilde{\mathbf{u}}}) \quad (60)$$

for which a turbulence closure model is assumed, e.g. eddy viscosity model or the more involved $k - \varepsilon$ [178] or Spalart-Allmaras models [475]. In Eq. (60) $\nabla^s \bar{\mathbf{u}}$ is the average deviatoric strain-rate tensor. The framework in Eq. (60) gives the two commonly used models: the Reynolds-Averaged Navier Stokes (RANS) model, best for steady flows [476, 259], and the Large Eddy Simulations (LES) model, using a sub-grid-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 [587] 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 [539]. Results of using ML to improve accuracy of closure models are, for example given in [518, 523]. 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 through inserting NN corrections which learn the scales evolution relations, greatly accelerating the computations by allowing lower resolution [280].

4.2.2 Shock dynamics

More accurate and faster shock-capturing by NN has been pursued in [481], 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.

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 [451, 218] or more general POD [52, 34, 435], PGD [136, 102], PCA [35],

SVD [327, 67]. Autoencoders employing different NN types [286, 371, 553, 342] and other nonlinear extensions of the previous techniques are a widely used approach for dealing with nonlinear cases typical in fluid dynamics [189]. These techniques also frequently include physics information to guarantee consistency [145].

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) [148]. ML is applied for the primal identification of structural systems (SSI) [471, 22], in particular of complex or historical structures, to assess their general and seismic vulnerability [438, 549] and facilitate ulterior health monitoring [357]. Feature extraction and model reduction is fundamental in these approaches [431]. Other areas where ML is employed is in the control of structures (e.g. active Tuned Mass Dampers [571, 109, 146]) under wind, seismic or crowd actions, or in structural design [222, 486, 228, 570]. We also comment in this section on the development of novel ML approaches based on ideas used in structural and finite element analyses.

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, Spain, etc) [131]. It is also a problem in the assessment of modern structures, since modeling assumptions may not have been sufficiently accurate [501]. 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 [196], 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 [492, 579]. K-Means and KNNs are also frequently used in SHM. For example, in [447] anomaly detection is performed using the (squared) “distance” $(\mathbf{x} - \bar{\mathbf{x}})S_k^{-1}(\mathbf{x} - \bar{\mathbf{x}})$ 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 [467]. Health monitoring of bridges is the focus in rather early research (e.g. the simple case analyzed in [319] through NNs). The traffic load [302] and ambient vibrations [36] 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 [279]. Basically, all types of bridges have been studied using ML techniques, namely steel girder bridges [384],

reinforced concrete T-bridges [208], cable stayed [29] and long suspension bridges [382], truss bridges [347] and arch bridges [247]. Different types of NN are used (e.g. Bayesian [28, 307, 383], Convolutional [379, 406], Recurrent [354, 353]), and the use of other techniques is also frequent as SVM, see for example [15, 568]

Apart from bridges and multi-story buildings [190, 520] 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 [310], where a CNN is used to extract features and a bi-directional gated RNN is employed to perform transfer learning from long-term dependencies. Similar works addressing SHM of dams are given [569, 459]. A review may be found in [439].

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 [263, 414, 95, 213] to monitor radial and lateral displacements in arch dams. Several ML techniques as Random Forest (RF), Boosted Regression Trees (BRT), NN, SVM and MARS are compared in [440] 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 [441]. 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 [207], 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 [268] with three classical methods (see review of Multicriteria Decision Making (MCDM) in [68]) 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 behaviour often depends on qualitative observations; it is therefore another field where machine learning techniques will have an important impact in the future [243].

Important types of structures considered in SHM are also aerogenerators or Wind Turbines (WT), see review in [105]. Here, two main components are typically analyzed: the blades and the gearbox [525]. SVM is a frequent ML technique used and acoustic noise is a source of relevant data for blades monitoring [420]. Deep NNs are also frequently employed when multiple sources of data are available, in particular CNNs are used to deal with images from drones [462, 199]. 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 [290]. 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 [251] 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 [250] 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, kNN 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 [309, 310]. Systems, 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 [330]. 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 [337] several ML methods as Naïve Bayes, kNN, 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) as AdaBoost (Adaptative 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 [360].

In SHM, the integration of data from different types or families of sensors (data fusion) is an important topic. Data fusion [205] brings challenges in SHM, but also the possibility of more accurate, integral prediction of the health of the structure [542]. For example, in [513] a data fusion system based on kNN 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) [225]. 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) [446, 419] is currently being performed using ML techniques. A recent review of these image-based techniques can be found in [132]. Another recent review of ML applications of SHM of civil structures can be found in [156].

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 [285], but which may result in significant losses. This issue is also of special relevancy in the aircraft industry [13]

4.3.2 Structural design and topology optimization

The design of components and structures is based on creativity and experience [333], so it is also an optimal field for the use of ML procedures, e.g. [7]. 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 [163] 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 [90]. Different structural systems are conceptually designed with the aid of ML techniques, including shear walls to sustain earthquakes (e.g. using GAN, in [329, 586]), shell structures [490, 589] and even the architectural volume [91]. A study and proposal of different ML techniques in architectural design can be found in [491].

Of course, one of the main disciplines in structural design is Topology Optimization (TO), and ML approaches (a combination coined “learning topology” in [365]) can be used to develop more robust schemes [99, 370] through tuning numerical parameters [332]. For example in [370] 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 [5]. For obtaining optimum thermal structures, GANs have been used to develop non-iterative structural TO [306]. Using ML to develop a non-iterative TO approach has also been addressed in [567]. A recent review of ML techniques in TO can be found in [369].

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 [44]. In [253] 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 [400]. 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 [83] several possibilities are addressed and applied to nonlinear truss structures and a (nonlinear) hyperelastic perforated plane strain structure. A similar approach is used in [557] for composite shells employing physics-based NNs. In [256] 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 .

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 [18] a system is developed with the aim of replicating human physiology. In [19] a similar approach is developed for nanofluid flow, while [422] studies hydrogen production.

In the field of multiphysics problems there exists a particularly appealing approach to machine learning; namely, that of port-Hamiltonian formalisms [509]. 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

$$\dot{\mathbf{z}} = \mathbf{J}\nabla\mathbf{H} + \mathbf{F}, \quad (61)$$

where \mathbf{J} is the classical (skew-symmetric) symplectic matrix, \mathbf{H} is the Hamiltonian (total energy of the system) and \mathbf{F} is the forcing term, which links the port-Hamiltonian system to other sub-systems. 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 fulfilment 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 [125], see also [340, 140, 341, 477, 363]. A recent review on the progress of these techniques can be found in [98].

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.

4.6.1 Classical manufacturing

In manufacturing, plasticity plays a fundamental role. Machine learning approaches to solve inelastic problems have already been addressed in Section 4.1.2 above. Research activities prior to 1965 are compiled in an interesting review by Monostori et al. [362]. More recently, another review compiled the works in different research endeavors within the field of manufacturing [401].

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 [409]. In [461] valuable research is reported in which Natural Language Processing (NLP) was applied to documentation from 2005 to 2027 in the field of smart manufacturing. The survey analyzes aspects ranging from decision support (prior to the moment a piece is manufactured), plant and operations health management (for the manufacturing process itself), data management, as a consequence of the vast amount of information produced by IoT (internet of Things) 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 [8] a review is given of literature that employs ML to perform diagnosis of manufacturing processes.

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 [418, 350, 252, 519]. 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 micro-scale, 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 [42, 575].

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, etc.). Both the design using AM and the error correction or compensation [387] 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 [120], for detecting geometrical defects in SLM-made components [242, 184], for building process maps relating variables to desired properties (e.g. as low porosity) [27], for predicting surface roughness in terms of process features [538], etc. 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 [81], to build process maps and optimize efficiency and surface roughness in SLM [581], to minimize support wasted material (optimize supports in a piece) in FDM [249], to predict and optimize resulting mechanical properties of the printed material [303] like strength (e.g., using CNN from thermal histories in [548] or FFNN in [46]), bending stiffness in AM composites [376] and stress-strain curves of binary composites using a combination of CNN and PCA [560].

NNs have also been used to create surrogate models with the purpose of mimicking the acoustic properties of AM replicas of a Stradivarius violin [498]. Reviews of techniques and different applications of machine learning in additive manufacturing may be found in [519, 119, 350, 405, 545, 209]. The review in [200] addresses in some detail physics-based proposals.

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 [171].

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 [177, 267], see the review of ML contributions in [135], see also [506] specifically 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) [191] are used to explore aerodynamic shapes [97]. 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 [94]. Further, ML is employed in the evaluation of design concepts from many candidates based on human preferences expressed in previous concepts [82]. There are also ML works that aid in the development of detailed and consistent CAD drawings from hand sketches [457], 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 [565], 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) are 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 [463]. The combination of topology optimization and generative design can be found in many endeavors [385, 41].

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 [104] and manufacturing processes [312] of new designs may be inferred from previous similar designs [576]. Indeed, many works focus on procedures to reuse parts of CAD schemes for electronic circuits [64] or to develop microfluidic devices [326, 503].

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 many problem solutions, robustness of the solution is important, 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, are also gratefully acknowledged by EC.

References

1. Google cloud: AI and machine learning products. innovative machine learning products and services on a trusted platform. URL <https://cloud.google.com/products/ai>
2. Build the most powerful models with C++. opennn.net, <https://www.opennn.net> (2022). URL <https://www.opennn.net/>
3. Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., Fieguth, P., Cao, X., Khosravi, A., Acharya, U.R., et al.: A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion* **76**, 243–297 (2021)
4. Abdi, H., Williams, L.J.: Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics* **2**(4), 433–459 (2010)
5. Abueidda, D.W., Koric, S., Sobh, N.A.: Topology optimization of 2d structures with nonlinearities using deep learning. *Computers & Structures* **237**, 106283 (2020)
6. Abueidda, D.W., Koric, S., Sobh, N.A., Sehitoglu, H.: Deep learning for plasticity and thermo-viscoplasticity. *International Journal of Plasticity* **136**, 102852 (2021)
7. Adeli, H., Yeh, C.: Explanation-based machine learning in engineering design. *Engineering Applications of Artificial Intelligence* **3**(2), 127–137 (1990)
8. Ademujimi, T.T., Brundage, M.P., Prabhu, V.V.: A review of current machine learning techniques used in manufacturing diagnosis. In: *IFIP International Conference on Advances in Production Management Systems*, pp. 407–415. Springer (2017)
9. Adriaans, P., Zantinge, D.: *Data mining*. Addison-Wesley Longman Publishing Co., Inc. (1997)
10. Aggarwal, A., Mittal, M., Battineni, G.: Generative adversarial network: An overview of theory and applications. *International Journal of Information Management Data Insights* **1**(1), 100004 (2021)
11. Agrawal, A., Choudhary, A.: Perspective: Materials informatics and big data: Realization of the “fourth paradigm” of science in materials science. *Apl Materials* **4**(5), 053208 (2016)
12. Ahmed, E., Jones, M., Marks, T.K.: An improved deep learning architecture for person re-identification. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3908–3916 (2015)
13. Ahmed, O., Wang, X., Tran, M.V., Ismadi, M.Z.: Advancements in fiber-reinforced polymer composite materials damage detection methods: towards achieving energy-efficient shm systems. *Composites Part B: Engineering* **223**, 109136 (2021)
14. Akinyelu, A.A.: Advances in spam detection for email spam, web spam, social network spam, and review spam: ML-based and nature-inspired-based techniques. *Journal of Computer Security* **29**(5), 473–529 (2021)
15. Alamdari, M.M., Rakotoarivelo, T., Khoa, N.L.D.: A spectral-based clustering for structural health monitoring of the sydney harbour bridge. *Mechanical Systems and Signal Processing* **87**, 384–400 (2017)
16. Alber, M., Buganza Tepole, A., Cannon, W.R., De, S., Dura-Bernal, S., Garikipati, K., Karniadakis, G., Lytton, W.W., Perdikaris, P., Petzold, L., et al.: Integrating machine learning and

- multiscale modeling—perspectives, challenges, and opportunities in the biological, biomedical, and behavioral sciences. *NPJ Digital Medicine* **2**(1), 1–11 (2019)
17. Aletti, M., Bortolossi, A., Perotto, S., Veneziani, A.: One-dimensional surrogate models for advection-diffusion problems. In: *Numerical Mathematics and Advanced Applications-ENUMATH 2013*, pp. 447–455. Springer (2015)
 18. Alexiadis, A.: Deep multiphysics: Coupling discrete multiphysics with machine learning to attain self-learning in-silico models replicating human physiology. *Artificial Intelligence in Medicine* **98**, 27–34 (2019)
 19. Alizadeh, R., Abad, J.M.N., Ameri, A., Mohebbi, M.R., Mehdizadeh, A., Zhao, D., Karimi, N.: 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. *Journal of the Taiwan Institute of Chemical Engineers* **124**, 290–306 (2021)
 20. Alizadeh, R., Allen, J.K., Mistree, F.: Managing computational complexity using surrogate models: a critical review. *Research in Engineering Design* **31**(3), 275–298 (2020)
 21. Alvarez, M.A., Rosasco, L., Lawrence, N.D., et al.: Kernels for vector-valued functions: A review. *Foundations and Trends® in Machine Learning* **4**(3), 195–266 (2012)
 22. Amezquita-Sanchez, J., Valtierra-Rodriguez, M., Adeli, H.: Machine learning in structural engineering. *Scientia Iranica* **27**(6), 2645–2656 (2020)
 23. Amores, V.J., Benítez, J.M., Montáns, F.J.: Data-driven, structure-based hyperelastic manifolds: A macro-micro-macro approach to reverse-engineer the chain behavior and perform efficient simulations of polymers. *Computers & Structures* **231**, 106209 (2020)
 24. Amores, V.J., Nguyen, K., Montáns, F.J.: On the network orientational affinity assumption in polymers and the micro–macro connection through the chain stretch. *Journal of the Mechanics and Physics of Solids* **148**, 104279 (2021)
 25. Amores, V.J., San Millan, F.J., Ben-Yelun, I., Montans, F.J.: A finite strain non-parametric hyperelastic extension of the classical phenomenological theory for orthotropic compressible composites. *Composites Part B: Engineering* **212**, 108591 (2021)
 26. Angelov, S., Stoimenova, E.: Cross-validated sequentially constructed multiple regression. In: *Annual Meeting of the Bulgarian Section of SIAM*, pp. 13–22. Springer (2017)
 27. Aoyagi, K., Wang, H., Sudo, H., Chiba, A.: Simple method to construct process maps for additive manufacturing using a support vector machine. *Additive Manufacturing* **27**, 353–362 (2019)
 28. Arangio, S., Beck, J.: Bayesian neural networks for bridge integrity assessment. *Structural Control and Health Monitoring* **19**(1), 3–21 (2012)
 29. Arangio, S., Bontempi, F.: Structural health monitoring of a cable-stayed bridge with bayesian neural networks. *Structure and Infrastructure Engineering* **11**(4), 575–587 (2015)
 30. Arbabi, H., Bunder, J.E., Samaey, G., Roberts, A.J., Kevrekidis, I.G.: Linking machine learning with multiscale numerics: Data-driven discovery of homogenized equations. *JOM* **72**(12), 4444–4457 (2020)
 31. Arrieta, A.B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., et al.: Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion* **58**, 82–115 (2020)
 32. Asgari, S., MirhoseiniNejad, S., Moazamigoodarzi, H., Gupta, R., Zheng, R., Puri, I.K.: A gray-box model for real-time transient temperature predictions in data centers. *Applied Thermal Engineering* **185**, 116319 (2021)
 33. Ashtiani, M.N., Raahemi, B.: Intelligent fraud detection in financial statements using machine learning and data mining: a systematic literature review. *IEEE Access* **10**, 72504–72525 (2021)
 34. Aubry, N.: On the hidden beauty of the proper orthogonal decomposition. *Theoretical and Computational Fluid Dynamics* **2**(5), 339–352 (1991)
 35. Audouze, C., De Vuyst, F., Nair, P.: Reduced-order modeling of parameterized PDEs using time–space-parameter principal component analysis. *International Journal for Numerical Methods in Engineering* **80**(8), 1025–1057 (2009)

36. Avci, O., Abdeljaber, O., Kiranyaz, S., Hussein, M., Gabbouj, M., Inman, D.J.: A review of vibration-based damage detection in civil structures: From traditional methods to machine learning and deep learning applications. *Mechanical Systems and Signal Processing* **147**, 107077 (2021)
37. Ayensa Jiménez, J.: 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 (2022)
38. Baklacioglu, T., Turan, O., Aydin, H.: Metaheuristics optimized machine learning modelling of environmental exergo-emissions for an aero-engine. *International Journal of Turbo & Jet-Engines* **39**(3), 411–426 (2019)
39. Balakrishnama, S., Ganapathiraju, A.: Linear discriminant analysis-a brief tutorial. *Institute for Signal and Information Processing* **18**(1998), 1–8 (1998)
40. Bank, D., Koenigstein, N., Giryas, R.: Autoencoders. *arXiv preprint arXiv:2003.05991* (2020)
41. Barbieri, L., Muzzupappa, M.: Performance-driven engineering design approaches based on generative design and topology optimization tools: A comparative study. *Applied Sciences* **12**(4), 2106 (2022)
42. Barchiesi, E., Spagnuolo, M., Placidi, L.: Mechanical metamaterials: a state of the art. *Mathematics and Mechanics of Solids* **24**(1), 212–234 (2019)
43. Bastek, J.H., Kumar, S., Telgen, B., Glaesener, R.N., Kochmann, D.M.: Inverting the structure–property map of truss metamaterials by deep learning. *Proceedings of the National Academy of Sciences* **119**(1) (2022)
44. Bathe, K.J.: *Finite element procedures*. Prentice Hall, 1996, 2nd edition KJ Bathe, Watertown, MA, 2014; also published by Higher Education Press China 2016. (2006)
45. Bathe, K.J., Wilson, E.L.: Solution methods for eigenvalue problems in structural mechanics. *International Journal for Numerical Methods in Engineering* **6**(2), 213–226 (1973)
46. Bayraktar, Ö., Uzun, G., Çakiroğlu, R., Guldaz, A.: Experimental study on the 3d-printed plastic parts and predicting the mechanical properties using artificial neural networks. *Polymers for Advanced Technologies* **28**(8), 1044–1051 (2017)
47. Becker, M., Teschner, M.: Weakly compressible SPH for free surface flows. In: *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 209–217 (2007)
48. Beel, J., Gipp, B.: Google scholar’s ranking algorithm: an introductory overview. In: *Proceedings of the 12th international conference on scientometrics and informetrics (ISSI’09)*, vol. 1, pp. 230–241. Rio de Janeiro (Brazil) (2009)
49. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation* **15**(6), 1373–1396 (2003)
50. Bengio, Y., Courville, A., Vincent, P.: Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **35**(8), 1798–1828 (2013)
51. Benítez, J.M., Montáns, F.J.: A simple and efficient numerical procedure to compute the inverse langevin function with high accuracy. *Journal of Non-Newtonian Fluid Mechanics* **261**, 153–163 (2018)
52. Berkooz, G., Holmes, P., Lumley, J.L.: The proper orthogonal decomposition in the analysis of turbulent flows. *Annual Review of Fluid Mechanics* **25**(1), 539–575 (1993)
53. Bertoldi, K., Vitelli, V., Christensen, J., Van Hecke, M.: Flexible mechanical metamaterials. *Nature Reviews Materials* **2**(11), 1–11 (2017)
54. Bezazi, A., Pierce, S.G., Worden, K., et al.: Fatigue life prediction of sandwich composite materials under flexural tests using a bayesian trained artificial neural network. *International Journal of Fatigue* **29**(4), 738–747 (2007)
55. Bhatti, G., Mohan, H., Singh, R.R.: Towards the future of smart electric vehicles: Digital twin technology. *Renewable and Sustainable Energy Reviews* **141**, 110801 (2021)
56. Bickel, S., Haider, P., Scheffer, T.: Learning to complete sentences. In: *European Conference on Machine Learning*, pp. 497–504. Springer (2005)

57. Bird, G.D., Gorrell, S.E., Salmon, J.L.: Dimensionality-reduction-based surrogate models for real-time design space exploration of a jet engine compressor blade. *Aerospace Science and Technology* **118**, 107077 (2021)
58. Bischl, B., Lang, M., Kotthoff, L., Schiffner, J., Richter, J., Studerus, E., Casalicchio, G., Jones, Z.M.: mlr: Machine learning in R. *The Journal of Machine Learning Research* **17**(1), 5938–5942 (2016)
59. Bishop, C.M.: Training with noise is equivalent to tikhonov regularization. *Neural Computation* **7**(1), 108–116 (1995)
60. Bisong, E.: Google cloud machine learning engine (cloud MLE). In: *Building Machine Learning and Deep Learning Models on Google Cloud Platform*, pp. 545–579. Springer (2019)
61. Bisong, E.: Numpy. In: *Building Machine Learning and Deep Learning Models on Google Cloud Platform*, pp. 91–113. Springer (2019)
62. Bock, F.E., Aydin, R.C., Cyron, C.J., Huber, N., Kalidindi, S.R., Klusemann, B.: A review of the application of machine learning and data mining approaches in continuum materials mechanics. *Frontiers in Materials* p. 110 (2019)
63. Bonet, J., Wood, R.D.: *Nonlinear continuum mechanics for finite element analysis*. Cambridge university press (1997)
64. Boning, D.S., Elfadel, I.A.M., Li, X.: A preliminary taxonomy for machine learning in vlsi cad. In: *Machine Learning in VLSI Computer-Aided Design*, pp. 1–16. Springer (2019)
65. Borkowski, L., Sorini, C., Chattopadhyay, A.: Recurrent neural network-based multiaxial plasticity model with regularization for physics-informed constraints. *Computers & Structures* **258**, 106678 (2022)
66. du Bos, M.L., Balabdaoui, F., Heidenreich, J.N.: Modeling stress-strain curves with neural networks: a scalable alternative to the return mapping algorithm. *Computational Materials Science* **178**, 109629 (2020)
67. Braconnier, T., Ferrier, M., Jouhaud, J.C., Montagnac, M., Sagaut, P.: Towards an adaptive pod/svd surrogate model for aeronautic design. *Computers & Fluids* **40**(1), 195–209 (2011)
68. de Brito, M.M., Evers, M.: Multi-criteria decision-making for flood risk management: a survey of the current state of the art. *Natural Hazards and Earth System Sciences* **16**(4), 1019–1033 (2016)
69. Bro, R., Smilde, A.K.: Principal component analysis. *Analytical Methods* **6**(9), 2812–2831 (2014)
70. Brodie, C.R., Constantin, A., Deen, R., Lukas, A.: Machine learning line bundle cohomology. *Fortschritte der Physik* **68**(1), 1900087 (2020)
71. Brunton, S.L., Kutz, J.N.: Methods for data-driven multiscale model discovery for materials. *Journal of Physics: Materials* **2**(4), 044002 (2019)
72. Brunton, S.L., Kutz, J.N.: *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press (2022)
73. Brunton, S.L., Proctor, J.L., Kutz, J.N.: Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences* **113**(15), 3932–3937 (2016)
74. Buizza, C., Casas, C.Q., Nadler, P., Mack, J., Marrone, S., Titus, Z., Le Cornec, C., Heylen, E., Dur, T., Ruiz, L.B., et al.: Data learning: Integrating data assimilation and machine learning. *Journal of Computational Science* **58**, 101525 (2022)
75. Bukka, S.R., Magee, A.R., Jaiman, R.K.: Deep convolutional recurrent autoencoders for flow field prediction. In: *International Conference on Offshore Mechanics and Arctic Engineering*, vol. 84409, p. V008T08A005. American Society of Mechanical Engineers (2020)
76. Burkov, A.: *The hundred-page machine learning book*, vol. 1. Andriy Burkov Quebec City, QC, Canada (2019)
77. Burkov, A.: *Machine learning engineering*, vol. 1. True Positive Incorporated (2020)
78. Burov, A., Burova, O.: Development of digital twin for composite pressure vessel. In: *Journal of Physics: Conference Series*, vol. 1441, p. 012133. IOP Publishing (2020)
79. Buşoniu, L., de Bruin, T., Tolić, D., Kober, J., Palunko, I.: Reinforcement learning for control: Performance, stability, and deep approximators. *Annual Reviews in Control* **46**, 8–28 (2018)

80. Caccin, M., Li, Z., Kermode, J.R., De Vita, A.: A framework for machine-learning-augmented multiscale atomistic simulations on parallel supercomputers. *International Journal of Quantum Chemistry* **115**(16), 1129–1139 (2015)
81. Caiazzo, F., Caggiano, A.: Laser direct metal deposition of 2024 Al alloy: trace geometry prediction via machine learning. *Materials* **11**(3), 444 (2020)
82. Camburn, B., He, Y., Raviselvam, S., Luo, J., Wood, K.: Machine learning-based design concept evaluation. *Journal of Mechanical Design* **142**(3), 031113 (2020)
83. Capuano, G., Rimoli, J.J.: Smart finite elements: A novel machine learning application. *Computer Methods in Applied Mechanics and Engineering* **345**, 363–381 (2019)
84. Carleo, G., Cirac, I., Cranmer, K., Daudet, L., Schuld, M., Tishby, N., Vogt-Maranto, L., Zdeborová, L.: Machine learning and the physical sciences. *Reviews of Modern Physics* **91**(4), 045002 (2019)
85. Carrara, P., De Lorenzis, L., Stainier, L., Ortiz, M.: Data-driven fracture mechanics. *Computer Methods in Applied Mechanics and Engineering* **372**, 113390 (2020)
86. Cayton, L.: Algorithms for manifold learning. University of California at San Diego Technical Report **12**(1-17), 1 (2005)
87. Champaney, V., Chinesta, F., Cueto, E.: Engineering empowered by physics-based and data-driven hybrid models: A methodological overview. *International Journal of Material Forming* **15**(3), 1–14 (2022)
88. Champion, K., Lusch, B., Kutz, J.N., Brunton, S.L.: Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences* **116**(45), 22445–22451 (2019)
89. Chan, S., Elsheikh, A.H.: A machine learning approach for efficient uncertainty quantification using multiscale methods. *Journal of Computational Physics* **354**, 493–511 (2018)
90. Chang, K.H., Cheng, C.Y.: Learning to simulate and design for structural engineering. In: *International Conference on Machine Learning*, pp. 1426–1436. PMLR (2020)
91. Chang, K.H., Cheng, C.Y., Luo, J., Murata, S., Nourbakhsh, M., Tsuji, Y.: Buildinggan: Graph-conditioned architectural volumetric design generation. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11956–11965 (2021)
92. Chattopadhyay, A., Hassanzadeh, P., Subramanian, D.: 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 Processes in Geophysics* **27**(3), 373–389 (2020)
93. Chazal, F., Michel, B.: An introduction to topological data analysis: fundamental and practical aspects for data scientists. *Frontiers in Artificial Intelligence* **4** (2021)
94. Chen, L.: Data-driven and machine learning based design creativity. Ph.D. thesis, Imperial College London (2020)
95. Chen, S., Gu, C., Lin, C., Zhao, E., Song, J.: Safety monitoring model of a super-high concrete dam by using rbf neural network coupled with kernel principal component analysis. *Mathematical Problems in Engineering* **2018** (2018)
96. Chen, T., Chen, H.: Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE Transactions on Neural Networks* **6**(4), 911–917 (1995)
97. Chen, W., Chiu, K., Fuge, M.: Aerodynamic design optimization and shape exploration using generative adversarial networks. In: *AIAA Scitech 2019 Forum*, p. 2351 (2019)
98. Cherifi, K.: An overview on recent machine learning techniques for port hamiltonian systems. *Physica D: Nonlinear Phenomena* **411**, 132620 (2020)
99. Chi, H., Zhang, Y., Tang, T.L.E., Mirabella, L., Dalloro, L., Song, L., Paulino, G.H.: Universal machine learning for topology optimization. *Computer Methods in Applied Mechanics and Engineering* **375**, 112739 (2021)
100. Chinesta, F., Cueto, E., Abisset-Chavanne, E., Duval, J.L., Khaldi, F.E.: Virtual, digital and hybrid twins: a new paradigm in data-based engineering and engineered data. *Archives of Computational Methods in Engineering* **27**(1), 105–134 (2020)

101. Chinesta, F., Cueto, E., Grmela, M., Moya, B., Pavelka, M., Šípka, M.: Learning physics from data: a thermodynamic interpretation. In: Workshop on Joint Structures and Common Foundations of Statistical Physics, Information Geometry and Inference for Learning, pp. 276–297. Springer (2020)
102. Chinesta, F., Ladeveze, P., Cueto, E.: A short review on model order reduction based on proper generalized decomposition. *Archives of Computational Methods in Engineering* **18**(4), 395–404 (2011)
103. Choi, S.Y., Cha, D.: Unmanned aerial vehicles using machine learning for autonomous flight; state-of-the-art. *Advanced Robotics* **33**(6), 265–277 (2019)
104. Chu, C.H., Hsu, Y.C.: Similarity assessment of 3d mechanical components for design reuse. *Robotics and Computer-Integrated Manufacturing* **22**(4), 332–341 (2006)
105. Ciang, C.C., Lee, J.R., Bang, H.J.: Structural health monitoring for a wind turbine system: a review of damage detection methods. *Measurement Science and Technology* **19**(12), 122001 (2008)
106. Ciftci, K., Hackl, K.: Model-free data-driven simulation of inelastic materials using structured data sets, tangent space information and transition rules. *Computational Mechanics* pp. 1–11 (2022)
107. Clifton, A., Kilcher, L., Lundquist, J., Fleming, P.: Using machine learning to predict wind turbine power output. *Environmental Research Letters* **8**(2), 024009 (2013)
108. Coelho, M., Roehl, D., Bletzinger, K.U.: Material model based on NURBS response surfaces. *Applied Mathematical Modelling* **51**, 574–586 (2017)
109. Colherinhas, G.B., de Moraes, M.V., Shzu, M.A., Avila, S.M.: Optimal pendulum tuned mass damper design applied to high towers using genetic algorithms: Two-dof modeling. *International Journal of Structural Stability and Dynamics* **19**(10), 1950125 (2019)
110. Conti, S., Müller, S., Ortiz, M.: Data-driven problems in elasticity. *Archive for Rational Mechanics and Analysis* **229**(1), 79–123 (2018)
111. Conti, S., Müller, S., Ortiz, M.: Data-driven finite elasticity. *Archive for Rational Mechanics and Analysis* **237**(1), 1–33 (2020)
112. Cranmer, M., Greydanus, S., Hoyer, S., Battaglia, P., Spergel, D., Ho, S.: Lagrangian neural networks. *arXiv preprint arXiv:2003.04630* (2020)
113. Crawford, M., Khoshgoftaar, T.M., Prusa, J.D., Richter, A.N., Al Najada, H.: Survey of review spam detection using machine learning techniques. *Journal of Big Data* **2**(1), 1–24 (2015)
114. Crespo, J., Duncan, O., Alderson, A., Montáns, F.J.: Auxetic orthotropic materials: Numerical determination of a phenomenological spline-based stored density energy and its implementation for finite element analysis. *Computer Methods in Applied Mechanics and Engineering* **371**, 113300 (2020)
115. Crespo, J., Latorre, M., Montáns, F.J.: WYPIWYG hyperelasticity for isotropic, compressible materials. *Computational Mechanics* **59**(1), 73–92 (2017)
116. Crespo, J., Montáns, F.J.: A continuum approach for the large strain finite element analysis of auxetic materials. *International Journal of Mechanical Sciences* **135**, 441–457 (2018)
117. Crespo, J., Montáns, F.J.: General solution procedures to compute the stored energy density of conservative solids directly from experimental data. *International Journal of Engineering Science* **141**, 16–34 (2019)
118. De Jong, K.: Learning with genetic algorithms: An overview. *Machine Learning* **3**(2), 121–138 (1988)
119. DebRoy, T., Mukherjee, T., Wei, H., Elmer, J., Milewski, J.: Metallurgy, mechanistic models and machine learning in metal printing. *Nature Reviews Materials* **6**(1), 48–68 (2021)
120. Delli, U., Chang, S.: Automated process monitoring in 3d printing using supervised machine learning. *Procedia Manufacturing* **26**, 865–870 (2018)
121. Demo, N., Tezzele, M., Rozza, G.: Pydmd: Python dynamic mode decomposition. *Journal of Open Source Software* **3**(22), 530 (2018)
122. Dener, A., Miller, M.A., Churchill, R.M., Munson, T., Chang, C.S.: Training neural networks under physical constraints using a stochastic augmented Lagrangian approach. *arXiv preprint arXiv:2009.07330* (2020)

123. Deng, Z., He, C., Liu, Y., Kim, K.C.: Super-resolution reconstruction of turbulent velocity fields using a generative adversarial network-based artificial intelligence framework. *Physics of Fluids* **31**(12), 125111 (2019)
124. Denkena, B., Bergmann, B., Witt, M.: Material identification based on machine-learning algorithms for hybrid workpieces during cylindrical operations. *Journal of Intelligent Manufacturing* **30**(6), 2449–2456 (2019)
125. Desai, S.A., Mattheakis, M., Sondak, D., Protopapas, P., Roberts, S.J.: Port-hamiltonian neural networks for learning explicit time-dependent dynamical systems. *Physical Review E* **104**(3), 034312 (2021)
126. Dhanalaxmi, B.: Machine learning and its emergence in the modern world and its contribution to artificial intelligence. In: 2020 International Conference for Emerging Technology (INCET), pp. 1–4. IEEE (2020)
127. Di Leoni, P.C., Lu, L., Meneveau, C., Karniadakis, G., Zaki, T.A.: DeepONet prediction of linear instability waves in high-speed boundary layers. *arXiv preprint arXiv:2105.08697* (2021)
128. Dijkstra, E.W., et al.: A note on two problems in connexion with graphs. *Numerische Mathematik* **1**(1), 269–271 (1959)
129. Dillon, J.V., Langmore, I., Tran, D., Brevdo, E., Vasudevan, S., Moore, D., Patton, B., Alemi, A., Hoffman, M., Saurous, R.A.: Tensorflow distributions. *arXiv preprint arXiv:1711.10604* (2017)
130. Ding, S., Lin, L., Wang, G., Chao, H.: Deep feature learning with relative distance comparison for person re-identification. *Pattern Recognition* **48**(10), 2993–3003 (2015)
131. Domaneschi, M., Noori, A.Z., Pietropinto, M.V., Cimellaro, G.P.: Seismic vulnerability assessment of existing school buildings. *Computers & Structures* **248**, 106522 (2021)
132. Dong, C.Z., Catbas, F.N.: A review of computer vision-based structural health monitoring at local and global levels. *Structural Health Monitoring* **20**(2), 692–743 (2021)
133. Duarte, A.C., Roldan, F., Tubau, M., Escur, J., Pascual, S., Salvador, A., Mohedano, E., McGuinness, K., Torres, J., Giro-i Nieto, X.: WAV2PIX: Speech-conditioned face generation using generative adversarial networks. In: ICASSP, pp. 8633–8637 (2019)
134. Duarte, F.: 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] (2018). URL <https://www.bbc.com/mundo/noticias-42916502>
135. Duffy, A.H.: The "what" and "how" of learning in design. *IEEE Expert* **12**(3), 71–76 (1997)
136. Dumon, A., Allery, C., Ammar, A.: Proper general decomposition (PGD) for the resolution of Navier–Stokes equations. *Journal of Computational Physics* **230**(4), 1387–1407 (2011)
137. Eggersmann, R., Kirchdoerfer, T., Reese, S., Stainier, L., Ortiz, M.: Model-free data-driven inelasticity. *Computer Methods in Applied Mechanics and Engineering* **350**, 81–99 (2019)
138. Eggersmann, R., Stainier, L., Ortiz, M., Reese, S.: Efficient data structures for model-free data-driven computational mechanics. *Computer Methods in Applied Mechanics and Engineering* **382**, 113855 (2021)
139. Eggersmann, R., Stainier, L., Ortiz, M., Reese, S.: Model-free data-driven computational mechanics enhanced by tensor voting. *Computer Methods in Applied Mechanics and Engineering* **373**, 113499 (2021)
140. Eidnes, S., Stasik, A.J., Sterud, C., Bøhn, E., Riemer-Sørensen, S.: Port-hamiltonian neural networks with state dependent ports. *arXiv preprint arXiv:2206.02660* (2022)
141. Eilers, P.H., Marx, B.D.: Flexible smoothing with b-splines and penalties. *Statistical science* **11**(2), 89–121 (1996)
142. El Kadi, H.: Modeling the mechanical behavior of fiber-reinforced polymeric composite materials using artificial neural networks—a review. *Composite Structures* **73**(1), 1–23 (2006)
143. El Said, B., Hallett, S.R.: Multiscale surrogate modelling of the elastic response of thick composite structures with embedded defects and features. *Composite Structures* **200**, 781–798 (2018)
144. Erchiqui, F., Kandil, N.: Neuronal networks approach for characterization of softened polymers. *Journal of Reinforced Plastics and Composites* **25**(5), 463–473 (2006)

145. Erichson, N.B., Muehlebach, M., Mahoney, M.W.: Physics-informed autoencoders for lyapunov-stable fluid flow prediction. arXiv preprint arXiv:1905.10866 (2019)
146. Etedali, S., Mollayi, N.: Cuckoo search-based least squares support vector machine models for optimum tuning of tuned mass dampers. *International Journal of Structural Stability and Dynamics* **18**(02), 1850028 (2018)
147. Eubank, R.L.: Nonparametric regression and spline smoothing. CRC press (1999)
148. Farrar, C.R., Worden, K.: Structural health monitoring: a machine learning perspective. John Wiley & Sons (2012)
149. Feng, N., Zhang, G., Khandelwal, K.: Finite strain FE2 analysis with data-driven homogenization using deep neural networks. *Computers & Structures* **263**, 106742 (2022)
150. Fernández, J., Chiachío, M., Chiachío, J., Muñoz, R., Herrera, F.: Uncertainty quantification in neural networks by approximate Bayesian computation: Application to fatigue in composite materials. *Engineering Applications of Artificial Intelligence* **107**, 104511 (2022)
151. Fernández, M., Fritzen, F., Weeger, O.: Material modeling for parametric, anisotropic finite strain hyperelasticity based on machine learning with application in optimization of metamaterials. *International Journal for Numerical Methods in Engineering* (2022)
152. Field, D., Ammouche, Y., Peña, J.M., Jérusalem, A.: Machine learning based multiscale calibration of mesoscopic constitutive models for composite materials: application to brain white matter. *Computational Mechanics* **67**(6), 1629–1643 (2021)
153. Fischer, C.C., Tibbetts, K.J., Morgan, D., Ceder, G.: Predicting crystal structure by merging data mining with quantum mechanics. *Nature Materials* **5**(8), 641–646 (2006)
154. Fish, J., Wagner, G.J., Keten, S.: Mesoscopic and multiscale modelling in materials. *Nature Materials* **20**(6), 774–786 (2021)
155. Fisher, R.A.: The use of multiple measurements in taxonomic problems. *Annals of Eugenics* **7**(2), 179–188 (1936)
156. Flah, M., Nunez, I., Ben Chaabene, W., Nehdi, M.L.: Machine learning algorithms in civil structural health monitoring: a systematic review. *Archives of Computational Methods in Engineering* **28**(4), 2621–2643 (2021)
157. Flaschel, M., Kumar, S., De Lorenzis, L.: Unsupervised discovery of interpretable hyperelastic constitutive laws. *Computer Methods in Applied Mechanics and Engineering* **381**, 113852 (2021)
158. Flaschel, M., Kumar, S., De Lorenzis, L.: Discovering plasticity models without stress data. *NPJ Computational Materials* **8**(1), 1–10 (2022)
159. Floyd, R.W.: Algorithm 97: shortest path. *Communications of the ACM* **5**(6), 345 (1962)
160. Frank, M., Drikakis, D., Charissis, V.: Machine-learning methods for computational science and engineering. *Computation* **8**(1), 15 (2020)
161. Frankel, A., Hamel, C.M., Bolintineanu, D., Long, K., Kramer, S.: Machine learning constitutive models of elastomeric foams. *Computer Methods in Applied Mechanics and Engineering* **391**, 114492 (2022)
162. Frankel, A.L., Safta, C., Alleman, C., Jones, R.: Mesh-based graph convolutional neural networks for modeling materials with microstructure. *Journal of Machine Learning for Modeling and Computing* **3**(1) (2022)
163. Freischlad, M., Schnellenbach-Held, M.: A machine learning approach for the support of preliminary structural design. *Advanced Engineering Informatics* **19**(4), 281–287 (2005)
164. Friedman, J.H.: Regularized discriminant analysis. *Journal of the American Statistical Association* **84**(405), 165–175 (1989)
165. Fu, K., Li, J., Zhang, Y., Shen, H., Tian, Y.: Model-guided multi-path knowledge aggregation for aerial saliency prediction. *IEEE Transactions on Image Processing* **29**, 7117–7127 (2020)
166. Fuchs, A., Heider, Y., Wang, K., Sun, W., Kaliske, M.: DNN2: A hyper-parameter reinforcement learning game for self-design of neural network based elasto-plastic constitutive descriptions. *Computers & Structures* **249**, 106505 (2021)
167. Fuhg, J.N., Böhm, C., Bouklas, N., Fau, A., Wriggers, P., Marino, M.: Model-data-driven constitutive responses: application to a multiscale computational framework. *International Journal of Engineering Science* **167**, 103522 (2021)

168. Fuhg, J.N., Bouklas, N., Jones, R.E.: Learning hyperelastic anisotropy from data via a tensor basis neural network. *arXiv preprint arXiv:2204.04529* (2022)
169. Fuhg, J.N., Fau, A., Bouklas, N., Marino, M.: Elasto-plasticity with convex model-data-driven yield functions (2022)
170. Gabel, J., Desaphy, J., Rognan, D.: Beware of machine learning-based scoring functions. on the danger of developing black boxes. *Journal of Chemical Information and Modeling* **54**(10), 2807–2815 (2014)
171. Ganin, Y., Bartunov, S., Li, Y., Keller, E., Saliceti, S.: Computer-aided design as language. *Advances in Neural Information Processing Systems* **34**, 5885–5897 (2021)
172. Gannouni, S., Maad, R.B.: Numerical analysis of smoke dispersion against the wind in a tunnel fire. *Journal of Wind Engineering and Industrial Aerodynamics* **158**, 61–68 (2016)
173. Gao, K., Mei, G., Piccialli, F., Cuomo, S., Tu, J., Huo, Z.: Julia language in machine learning: Algorithms, applications, and open issues. *Computer Science Review* **37**, 100254 (2020)
174. Garg, A., Panigrahi, B.K.: Multi-dimensional digital twin of energy storage system for electric vehicles: A brief review. *Energy Storage* **3**(6), e242 (2021)
175. Garg, S., Gupta, H., Chakraborty, S.: Assessment of deepoNet for time dependent reliability analysis of dynamical systems subjected to stochastic loading. *Engineering Structures* **270**, 114811 (2022)
176. Gaurav, D., Tiwari, S.M., Goyal, A., Gandhi, N., Abraham, A.: Machine intelligence-based algorithms for spam filtering on document labeling. *Soft Computing* **24**(13), 9625–9638 (2020)
177. Gero, J.S.: Creativity, emergence and evolution in design. *Knowledge-Based Systems* **9**(7), 435–448 (1996)
178. Gerolymos, G., Vallet, I.: Implicit computation of three-dimensional compressible Navier-Stokes equations using k-epsilon closure. *AIAA Journal* **34**(7), 1321–1330 (1996)
179. Ghosh, A., SahaRay, R., Chakrabarty, S., Bhadra, S.: Robust generalised quadratic discriminant analysis. *Pattern Recognition* **117**, 107981 (2021)
180. Ghoting, A., Krishnamurthy, R., Pednault, E., Reinwald, B., Sindhvani, V., Tatikonda, S., Tian, Y., Vaithyanathan, S.: SystemML: Declarative machine learning on mapreduce. In: 2011 IEEE 27th International Conference on Data Engineering, pp. 231–242. IEEE (2011)
181. Giacinto, G., Paolucci, R., Roli, F.: Application of neural networks and statistical pattern recognition algorithms to earthquake risk evaluation. *Pattern Recognition Letters* **18**(11–13), 1353–1362 (1997)
182. Gin, C.R., Shea, D.E., Brunton, S.L., Kutz, J.N.: DeepGreen: deep learning of Green's functions for nonlinear boundary value problems. *Scientific Reports* **11**(1), 1–14 (2021)
183. Glaessgen, E., Stargel, D.: 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 (2012)
184. Gobert, C., Reutzel, E.W., Petrich, J., Nassar, A.R., Phoha, S.: Application of supervised machine learning for defect detection during metallic powder bed fusion additive manufacturing using high resolution imaging. *Additive Manufacturing* **21**, 517–528 (2018)
185. González, D., Chinesta, F., Cueto, E.: Learning corrections for hyperelastic models from data. *Frontiers in Materials* **6**, 14 (2019)
186. González, D., Chinesta, F., Cueto, E.: Thermodynamically consistent data-driven computational mechanics. *Continuum Mechanics and Thermodynamics* **31**(1), 239–253 (2019)
187. González, D., Chinesta, F., Cueto, E.: Learning non-markovian physics from data. *Journal of Computational Physics* **428**, 109982 (2021)
188. González, D., García-González, A., Chinesta, F., Cueto, E.: A data-driven learning method for constitutive modeling: Application to vascular hyperelastic soft tissues. *Materials* **13**(10), 2319 (2020)
189. Gonzalez, F.J., Balajewicz, M.: Deep convolutional recurrent autoencoders for learning low-dimensional feature dynamics of fluid systems. *arXiv preprint arXiv:1808.01346* (2018)
190. González, M.P., Zapico, J.L.: Seismic damage identification in buildings using neural networks and modal data. *Computers & Structures* **86**(3–5), 416–426 (2008)

191. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. *Communications of the ACM* **63**(11), 139–144 (2020)
192. Goswami, S., Anitescu, C., Chakraborty, S., Rabczuk, T.: Transfer learning enhanced physics informed neural network for phase-field modeling of fracture. *Theoretical and Applied Fracture Mechanics* **106**, 102447 (2020)
193. Goswami, S., Yin, M., Yu, Y., Karniadakis, G.E.: A physics-informed variational DeepONet for predicting crack path in quasi-brittle materials. *Computer Methods in Applied Mechanics and Engineering* **391**, 114587 (2022)
194. Grefenstette, J.J.: Genetic algorithms and machine learning. In: *Proceedings of the Sixth Annual Conference on Computational Learning Theory*, pp. 3–4 (1993)
195. Greydanus, S., Dzamba, M., Yosinski, J.: Hamiltonian neural networks. *Advances in Neural Information Processing Systems* **32** (2019)
196. Gui, G., Pan, H., Lin, Z., Li, Y., Yuan, Z.: Data-driven support vector machine with optimization techniques for structural health monitoring and damage detection. *KSCE Journal of Civil Engineering* **21**(2), 523–534 (2017)
197. Guilleminot, J., Dolbow, J.E.: Data-driven enhancement of fracture paths in random composites. *Mechanics Research Communications* **103**, 103443 (2020)
198. Gulli, A., Pal, S.: *Deep learning with Keras*. Packt Publishing Ltd (2017)
199. Guo, J., Liu, C., Cao, J., Jiang, D.: Damage identification of wind turbine blades with deep convolutional neural networks. *Renewable Energy* **174**, 122–133 (2021)
200. Guo, S., Agarwal, M., Cooper, C., Tian, Q., Gao, R.X., Grace, W.G., Guo, Y.: Machine learning for metal additive manufacturing: Towards a physics-informed data-driven paradigm. *Journal of Manufacturing Systems* **62**, 145–163 (2022)
201. Hadden, C.M., Klimek-McDonald, D.R., Pineda, E.J., King, J.A., Reichenadter, A.M., Miskioglu, I., Gowtham, S., Odegard, G.M.: Mechanical properties of graphene nanoplatelet/carbon fiber/epoxy hybrid composites: Multiscale modeling and experiments. *Carbon* **95**, 100–112 (2015)
202. Haghighat, E., Juanes, R.: Sciann: A keras/tensorflow wrapper for scientific computations and physics-informed deep learning using artificial neural networks. *Computer Methods in Applied Mechanics and Engineering* **373**, 113552 (2021)
203. Haghighat, E., Raissi, M., Moure, A., Gomez, H., Juanes, R.: A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics. *Computer Methods in Applied Mechanics and Engineering* **379**, 113741 (2021)
204. Haik, W., Maday, Y., Chamoin, L.: 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* (2021)
205. Hall, D., Llinas, J.: *Multisensor data fusion*. CRC press (2001)
206. Hanifa, R.M., Isa, K., Mohamad, S.: A review on speaker recognition: Technology and challenges. *Computers & Electrical Engineering* **90**, 107005 (2021)
207. Hariri-Ardebili, M.A., Pourkamali-Anaraki, F.: Simplified reliability analysis of multi hazard risk in gravity dams via machine learning techniques. *Archives of Civil and Mechanical Engineering* **18**(2), 592–610 (2018)
208. Hasançebi, O., Dumlupınar, T.: Linear and nonlinear model updating of reinforced concrete t-beam bridges using artificial neural networks. *Computers & Structures* **119**, 1–11 (2013)
209. Hashemi, S.M., Parvizi, S., Baghbanijavid, H., Tan, A.T., Nematollahi, M., Ramazani, A., Fang, N.X., Elahinia, M.: Computational modelling of process–structure–property–performance relationships in metal additive manufacturing: A review. *International Materials Reviews* **67**(1), 1–46 (2022)
210. Hashemipour, S., Ali, M.: Amazon web services (AWS)—an overview of the on-demand cloud computing platform. In: *International Conference for Emerging Technologies in Computing*, pp. 40–47. Springer (2020)
211. Hassan, R.J., Abdulazeez, A.M., et al.: Deep learning convolutional neural network for face recognition: A review. *International Journal of Science and Business* **5**(2), 114–127 (2021)

212. Hastie, T., Tibshirani, R., Buja, A.: Flexible discriminant analysis by optimal scoring. *Journal of the American Statistical Association* **89**(428), 1255–1270 (1994)
213. He, Q., Gu, C., Valente, S., Zhao, E., Liu, X., Yuan, D.: Multi-arch dam safety evaluation based on statistical analysis and numerical simulation. *Scientific Reports* **12**(1), 1–19 (2022)
214. He, Q., Laurence, D.W., Lee, C.H., Chen, J.S.: Manifold learning based data-driven modeling for soft biological tissues. *Journal of Biomechanics* **117**, 110124 (2021)
215. He, S., Shin, H.S., Tsourdos, A.: Computational missile guidance: a deep reinforcement learning approach. *Journal of Aerospace Information Systems* **18**(8), 571–582 (2021)
216. He, X., He, Q., Chen, J.S.: Deep autoencoders for physics-constrained data-driven nonlinear materials modeling. *Computer Methods in Applied Mechanics and Engineering* **385**, 114034 (2021)
217. Hebb, D.O.: *The organization of behavior: A neuropsychological theory*. Psychology Press (2005)
218. Hemati, M.S., Williams, M.O., Rowley, C.W.: Dynamic mode decomposition for large and streaming datasets. *Physics of Fluids* **26**(11), 111701 (2014)
219. Hernández, Q., Badías, A., Chinesta, F., Cueto, E.: Thermodynamics-informed graph neural networks. *arXiv preprint arXiv:2203.01874* (2022)
220. Hernandez, Q., Badias, A., Gonzalez, D., Chinesta, F., Cueto, E.: Deep learning of thermodynamics-aware reduced-order models from data. *Computer Methods in Applied Mechanics and Engineering* **379**, 113763 (2021)
221. Hernández, Q., Badías, A., González, D., Chinesta, F., Cueto, E.: Structure-preserving neural networks. *Journal of Computational Physics* **426**, 109950 (2021)
222. Herrada, F., García-Martínez, J., Fraile, A., Hermanns, L., Montáns, F.: A method for performing efficient parametric dynamic analyses in large finite element models undergoing structural modifications. *Engineering Structures* **131**, 625–638 (2017)
223. Hershey, J.R., Olsen, P.A.: Approximating the Kullback Leibler divergence between Gaussian mixture models. In: 2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07, vol. 4, pp. IV–317. IEEE (2007)
224. Hkdh, B.: Neural networks in materials science. *ISIJ International* **39**(10), 966–979 (1999)
225. Ho, L.V., Nguyen, D.H., Mousavi, M., De Roeck, G., Bui-Tien, T., Gandomi, A.H., Wahab, M.A.: A hybrid computational intelligence approach for structural damage detection using marine predator algorithm and feedforward neural networks. *Computers & Structures* **252**, 106568 (2021)
226. Hofmann, T., Schölkopf, B., Smola, A.J.: Kernel methods in machine learning. *The Annals of Statistics* **36**(3), 1171–1220 (2008)
227. Hong, S.J., Chun, H., Lee, J., Kim, B.H., Seo, M.H., Kang, J., Han, B.: First-principles-based machine-learning molecular dynamics for crystalline polymers with van der waals interactions. *The Journal of Physical Chemistry Letters* **12**(25), 6000–6006 (2021)
228. Hong, T., Wang, Z., Luo, X., Zhang, W.: State-of-the-art on research and applications of machine learning in the building life cycle. *Energy and Buildings* **212**, 109831 (2020)
229. Hoshyar, A.N., Samali, B., Liyanapathirana, R., Houshyar, A.N., Yu, Y.: Structural damage detection and localization using a hybrid method and artificial intelligence techniques. *Structural Health Monitoring* **19**(5), 1507–1523 (2020)
230. Hosmer Jr, D.W., Lemeshow, S., Sturdivant, R.X.: *Applied logistic regression*, vol. 398. John Wiley & Sons (2013)
231. Hou, C., Wang, J., Wu, Y., Yi, D.: Local linear transformation embedding. *Neurocomputing* **72**(10–12), 2368–2378 (2009)
232. Huang, D., Fuhg, J.N., Weißenfels, C., Wriggers, P.: A machine learning based plasticity model using proper orthogonal decomposition. *Computer Methods in Applied Mechanics and Engineering* **365**, 113008 (2020)
233. Ibanez, R., Abisset-Chavanne, E., Aguado, J.V., Gonzalez, D., Cueto, E., Chinesta, F.: A manifold learning approach to data-driven computational elasticity and inelasticity. *Archives of Computational Methods in Engineering* **25**(1), 47–57 (2018)

- 234. Ibañez, R., Borzacchiello, D., Aguado, J.V., Abisset-Chavanne, E., Cueto, E., Ladeveze, P., Chinesta, F.: Data-driven non-linear elasticity: constitutive manifold construction and problem discretization. *Computational Mechanics* **60**(5), 813–826 (2017)
- 235. Ibanez, R., Gilormini, P., Cueto, E., Chinesta, F.: Numerical experiments on unsupervised manifold learning applied to mechanical modeling of materials and structures. *Comptes Rendus. Mécanique* **348**(10-11), 937–958 (2020)
- 236. Ibragimova, O., Brahme, A., Muhammad, W., Lévesque, J., Inal, K.: A new ANN based crystal plasticity model for fcc materials and its application to non-monotonic strain paths. *International Journal of Plasticity* **144**, 103059 (2021)
- 237. Ij, H.: Statistics versus machine learning. *Nature Methods* **15**(4), 233 (2018)
- 238. Innes, M.: Flux: Elegant machine learning with Julia. *Journal of Open Source Software* **3**(25), 602 (2018)
- 239. Innes, M., Edelman, A., Fischer, K., Rackauckas, C., Saba, E., Shah, V.B., Tebbutt, W.: A differentiable programming system to bridge machine learning and scientific computing. *arXiv preprint arXiv:1907.07587* (2019)
- 240. Inoue, H.: Data augmentation by pairing samples for images classification. *arXiv preprint arXiv:1801.02929* (2018)
- 241. Jackson, N.E., Webb, M.A., de Pablo, J.J.: Recent advances in machine learning towards multiscale soft materials design. *Current Opinion in Chemical Engineering* **23**, 106–114 (2019)
- 242. zur Jacobsmühlen, J., Kleszczynski, S., Witt, G., Merhof, D.: Detection of elevated regions in surface images from laser beam melting processes. In: *IECON 2015-41st Annual Conference of the IEEE Industrial Electronics Society*, pp. 001270–001275. IEEE (2015)
- 243. Jafari, M.: System identification of a soil tunnel based on a hybrid artificial neural network–numerical model approach. *Iranian Journal of Science and Technology, Transactions of Civil Engineering* **44**(3), 889–899 (2020)
- 244. Jagtap, A.D., Kharazmi, E., Karniadakis, G.E.: Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. *Computer Methods in Applied Mechanics and Engineering* **365**, 113028 (2020)
- 245. Jang, D.P., Fazily, P., Yoon, J.W.: Machine learning-based constitutive model for J2-plasticity. *International Journal of Plasticity* **138**, 102919 (2021)
- 246. Jansson, T., Nilsson, L., Redhe, M.: Using surrogate models and response surfaces in structural optimization—with application to crashworthiness design and sheet metal forming. *Structural and Multidisciplinary Optimization* **25**(2), 129–140 (2003)
- 247. Jayasundara, N., Thambiratnam, D., Chan, T., Nguyen, A.: Damage detection and quantification in deck type arch bridges using vibration based methods and artificial neural networks. *Engineering Failure Analysis* **109**, 104265 (2020)
- 248. Jha, D., Singh, S., Al-Bahrani, R., Liao, W.k., Choudhary, A., De Graef, M., Agrawal, A.: Extracting grain orientations from EBSD patterns of polycrystalline materials using convolutional neural networks. *Microscopy and Microanalysis* **24**(5), 497–502 (2018)
- 249. Jiang, J., Hu, G., Li, X., Xu, X., Zheng, P., Stringer, J.: Analysis and prediction of printable bridge length in fused deposition modelling based on back propagation neural network. *Virtual and Physical Prototyping* **14**(3), 253–266 (2019)
- 250. Jiménez, A.A., Márquez, F.P.G., Moraleda, V.B., Muñoz, C.Q.G.: Linear and nonlinear features and machine learning for wind turbine blade ice detection and diagnosis. *Renewable Energy* **132**, 1034–1048 (2019)
- 251. Jiménez, A.A., Zhang, L., Muñoz, C.Q.G., Márquez, F.P.G.: Maintenance management based on machine learning and nonlinear features in wind turbines. *Renewable Energy* **146**, 316–328 (2020)
- 252. Jin, Z., Zhang, Z., Demir, K., Gu, G.X.: Machine learning for advanced additive manufacturing. *Matter* **3**(5), 1541–1556 (2020)
- 253. Jokar, M., Semperlotti, F.: Finite element network analysis: A machine learning based computational framework for the simulation of physical systems. *Computers & Structures* **247**, 106484 (2021)

254. Jovanović, M.R., Schmid, P.J., Nichols, J.W.: Sparsity-promoting dynamic mode decomposition. *Physics of Fluids* **26**(2), 024103 (2014)
255. Jung, J., Yoon, J.I., Park, H.K., Jo, H., Kim, H.S.: Microstructure design using machine learning generated low dimensional and continuous design space. *Materialia* **11**, 100690 (2020)
256. Jung, J., Yoon, K., Lee, P.S.: Deep learned finite elements. *Computer Methods in Applied Mechanics and Engineering* **372**, 113401 (2020)
257. Kadic, M., Milton, G.W., van Hecke, M., Wegener, M.: 3d metamaterials. *Nature Reviews Physics* **1**(3), 198–210 (2019)
258. Kaehler, A., Bradski, G.: *Learning OpenCV 3: computer vision in C++ with the OpenCV library*. " O'Reilly Media, Inc." (2016)
259. Kalitzin, G., Medic, G., Iaccarino, G., Durbin, P.: Near-wall behavior of RANS turbulence models and implications for wall functions. *Journal of Computational Physics* **204**(1), 265–291 (2005)
260. Kamath, C.: On mining scientific datasets. In: *Data Mining for Scientific and Engineering Applications*, pp. 1–21. Springer (2001)
261. Kanno, Y.: Data-driven computing in elasticity via kernel regression. *Theoretical and Applied Mechanics Letters* **8**(6), 361–365 (2018)
262. Kanouté, P., Boso, D., Chaboche, J.L., Schrefler, B.: Multiscale methods for composites: a review. *Archives of Computational Methods in Engineering* **16**(1), 31–75 (2009)
263. Kao, C.Y., Loh, C.H.: Monitoring of long-term static deformation data of fei-tsui arch dam using artificial neural network-based approaches. *Structural Control and Health Monitoring* **20**(3), 282–303 (2013)
264. Karniadakis, G.E., Kevrekidis, I.G., Lu, L., Perdikaris, P., Wang, S., Yang, L.: Physics-informed machine learning. *Nature Reviews Physics* **3**(6), 422–440 (2021)
265. Karpatne, A., Atluri, G., Faghmous, J.H., Steinbach, M., Banerjee, A., Ganguly, A., Shekhar, S., Samatova, N., Kumar, V.: Theory-guided data science: A new paradigm for scientific discovery from data. *IEEE Transactions on Knowledge and Data Engineering* **29**(10), 2318–2331 (2017)
266. Kashinath, K., Mustafa, M., Albert, A., Wu, J., Jiang, C., Esmaeilzadeh, S., Azizadenesheli, K., Wang, R., Chattopadhyay, A., Singh, A., et al.: Physics-informed machine learning: case studies for weather and climate modelling. *Philosophical Transactions of the Royal Society A* **379**(2194), 20200093 (2021)
267. Khan, S., Awan, M.J.: A generative design technique for exploring shape variations. *Advanced Engineering Informatics* **38**, 712–724 (2018)
268. Khosravi, K., Shahabi, H., Pham, B.T., Adamowski, J., Shirzadi, A., Pradhan, B., Dou, J., Ly, H.B., Gróf, G., Ho, H.L., et al.: A comparative assessment of flood susceptibility modeling using multi-criteria decision-making analysis and machine learning methods. *Journal of Hydrology* **573**, 311–323 (2019)
269. Khurana, S., Saxena, S., Jain, S., Dixit, A.: Predictive modeling of engine emissions using machine learning: A review. *Materials Today: Proceedings* **38**, 280–284 (2021)
270. Kim, C., Batra, R., Chen, L., Tran, H., Ramprasad, R.: Polymer design using genetic algorithm and machine learning. *Computational Materials Science* **186**, 110067 (2021)
271. Kim, J.W., Lee, B.H., Shaw, M.J., Chang, H.L., Nelson, M.: Application of decision-tree induction techniques to personalized advertisements on internet storefronts. *International Journal of Electronic Commerce* **5**(3), 45–62 (2001)
272. Kim, P.: *Matlab deep learning: With machine learning, neural networks and artificial intelligence*. Apress (2017)
273. Kim, Y., Park, H.K., Jung, J., Asghari-Rad, P., Lee, S., Kim, J.Y., Jung, H.G., Kim, H.S.: Exploration of optimal microstructure and mechanical properties in continuous microstructure space using a variational autoencoder. *Materials & Design* **202**, 109544 (2021)
274. King, D.E.: Dlib-ml: A machine learning toolkit. *The Journal of Machine Learning Research* **10**, 1755–1758 (2009)
275. Kirchdoerfer, T., Ortiz, M.: Data-driven computational mechanics. *Computer Methods in Applied Mechanics and Engineering* **304**, 81–101 (2016)

276. Kirchdoerfer, T., Ortiz, M.: Data driven computing with noisy material data sets. *Computer Methods in Applied Mechanics and Engineering* **326**, 622–641 (2017)
277. Klein, D.K., Fernández, M., Martin, R.J., Neff, P., Weeger, O.: Polyconvex anisotropic hyperelasticity with neural networks. *Journal of the Mechanics and Physics of Solids* **159**, 104703 (2022)
278. Kleinbaum, D.G., Dietz, K., Gail, M., Klein, M., Klein, M.: *Logistic regression*. Springer (2002)
279. Ko, J., Ni, Y.Q.: Technology developments in structural health monitoring of large-scale bridges. *Engineering Structures* **27**(12), 1715–1725 (2005)
280. Kochkov, D., Smith, J.A., Alieva, A., Wang, Q., Brenner, M.P., Hoyer, S.: Machine learning–accelerated computational fluid dynamics. *Proceedings of the National Academy of Sciences* **118**(21), e2101784118 (2021)
281. Kolodziejczyk, F., Mortazavi, B., Rabczuk, T., Zhuang, X.: Machine learning assisted multi-scale modeling of composite phase change materials for li-ion batteries’ thermal management. *International Journal of Heat and Mass Transfer* **172**, 121199 (2021)
282. Kontolati, K., Alix-Williams, D., Boffi, N.M., Falk, M.L., Rycroft, C.H., Shields, M.D.: Manifold learning for coarse-graining atomistic simulations: Application to amorphous solids. *Acta Materialia* **215**, 117008 (2021)
283. Kossaifi, J., Panagakis, Y., Anandkumar, A., Pantic, M.: Tensorly: Tensor learning in python. *arXiv preprint arXiv:1610.09555* (2016)
284. Koumoulos, E., Konstantopoulos, G., Charitidis, C.: Applying machine learning to nanoindentation data of (nano-) enhanced composites. *Fibers* **8**(1), 3 (2019)
285. Kralovec, C., Schagerl, M.: Review of structural health monitoring methods regarding a multi-sensor approach for damage assessment of metal and composite structures. *Sensors* **20**(3), 826 (2020)
286. Kramer, M.A.: Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal* **37**(2), 233–243 (1991)
287. Krzeczowski, S.A.: Measurement of liquid droplet disintegration mechanisms. *International Journal of Multiphase Flow* **6**(3), 227–239 (1980)
288. Kulkarni, K.S., Kim, D.K., Sekar, S., Samui, P.: Model of least square support vector machine (lssvm) for prediction of fracture parameters of concrete. *International Journal of Concrete Structures and Materials* **5**(1), 29–33 (2011)
289. Ladevèze, P., Néron, D., Gerbaud, P.W.: Data-driven computation for history-dependent materials. *Comptes Rendus Mécanique* **347**(11), 831–844 (2019)
290. Laflamme, S., Cao, L., Chatzi, E., Ubertini, F.: Damage detection and localization from dense network of strain sensors. *Shock and Vibration* **2016** (2016)
291. Lakshminarayan, K., Harp, S.A., Goldman, R.P., Samad, T., et al.: Imputation of missing data using machine learning techniques. In: *KDD*, vol. 96 (1996)
292. Langley, P., et al.: The changing science of machine learning. *Machine learning* **82**(3), 275–279 (2011)
293. Lantz, B.: *Machine learning with R: expert techniques for predictive modeling*. Packt publishing ltd (2019)
294. Latorre, M., De Rosa, E., Montáns, F.J.: Understanding the need of the compression branch to characterize hyperelastic materials. *International Journal of Non-Linear Mechanics* **89**, 14–24 (2017)
295. Latorre, M., Mohammadkhah, M., Simms, C.K., Montáns, F.J.: A continuum model for tension-compression asymmetry in skeletal muscle. *Journal of the Mechanical Behavior of Biomedical Materials* **77**, 455–460 (2018)
296. Latorre, M., Montáns, F.J.: Extension of the Sussman–Bathe spline-based hyperelastic model to incompressible transversely isotropic materials. *Computers & Structures* **122**, 13–26 (2013)
297. Latorre, M., Montáns, F.J.: WYPiWYG hyperelasticity without inversion formula: Application to passive ventricular myocardium. *Computers & Structures* **185**, 47–58 (2017)
298. Latorre, M., Montáns, F.J.: Experimental data reduction for hyperelasticity. *Computers & Structures* **232**, 105919 (2020)

299. Latorre, M., Peña, E., Montáns, F.J.: Determination and finite element validation of the WYPIWYG strain energy of superficial fascia from experimental data. *Annals of Biomedical Engineering* **45**(3), 799–810 (2017)
300. Le, Q.V., Ngiam, J., Coates, A., Lahiri, A., Prochnow, B., Ng, A.Y.: On optimization methods for deep learning. In: ICML (2011)
301. Lee, D.W., Hong, S.H., Cho, S.S., Joo, W.S.: A study on fatigue damage modeling using neural networks. *Journal of Mechanical Science and Technology* **19**(7), 1393–1404 (2005)
302. Lee, J., Kim, J., Yun, C.B., Yi, J., Shim, J.: Health-monitoring method for bridges under ordinary traffic loadings. *Journal of Sound and Vibration* **257**(2), 247–264 (2002)
303. Lewandowski, J.J., Seifi, M.: Metal additive manufacturing: A review of mechanical properties (postprint) (2016)
304. Lewis, F.L., Liu, D.: Reinforcement learning and approximate dynamic programming for feedback control, vol. 17. John Wiley & Sons (2013)
305. Leygue, A., Coret, M., Réthoré, J., Stainier, L., Verron, E.: Data-based derivation of material response. *Computer Methods in Applied Mechanics and Engineering* **331**, 184–196 (2018)
306. Li, B., Huang, C., Li, X., Zheng, S., Hong, J.: Non-iterative structural topology optimization using deep learning. *Computer-Aided Design* **115**, 172–180 (2019)
307. Li, G., Liu, Q., Zhao, S., Qiao, W., Ren, X.: Automatic crack recognition for concrete bridges using a fully convolutional neural network and naive bayes data fusion based on a visual detection system. *Measurement Science and Technology* **31**(7), 075403 (2020)
308. Li, X., Roth, C.C., Mohr, D.: Machine-learning based temperature-and rate-dependent plasticity model: application to analysis of fracture experiments on dp steel. *International Journal of Plasticity* **118**, 320–344 (2019)
309. Li, Y., Bao, T., Chen, H., Zhang, K., Shu, X., Chen, Z., Hu, Y.: A large-scale sensor missing data imputation framework for dams using deep learning and transfer learning strategy. *Measurement* **178**, 109377 (2021)
310. Li, Y., Bao, T., Chen, Z., Gao, Z., Shu, X., Zhang, K.: A missing sensor measurement data reconstruction framework powered by multi-task gaussian process regression for dam structural health monitoring systems. *Measurement* **186**, 110085 (2021)
311. Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., Anandkumar, A.: Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895* (2020)
312. Li, Z., Zhou, X., Liu, W., Niu, Q., Kong, C.: A similarity-based reuse system for injection mold design in automotive interior industry. *The International Journal of Advanced Manufacturing Technology* **87**(5), 1783–1795 (2016)
313. Lin, X., Si, Z., Fu, W., Yang, J., Guo, S., Cao, Y., Zhang, J., Wang, X., Liu, P., Jiang, K., et al.: Intelligent identification of two-dimensional nanostructures by machine-learning optical microscopy. *Nano Research* **11**(12), 6316–6324 (2018)
314. Ling, J., Jones, R., Templeton, J.: Machine learning strategies for systems with invariance properties. *Journal of Computational Physics* **318**, 22–35 (2016)
315. Liu, B., Vu-Bac, N., Zhuang, X., Fu, X., Rabczuk, T.: Stochastic full-range multiscale modeling of thermal conductivity of polymeric carbon nanotubes composites: A machine learning approach. *Composite Structures* **289**, 115393 (2022)
316. Liu, H.H., Zhang, J., Liang, F., Temizel, C., Basri, M.A., Mesdour, R.: Incorporation of physics into machine learning for production prediction from unconventional reservoirs: A brief review of the gray-box approach. *SPE Reservoir Evaluation & Engineering* **24**(04), 847–858 (2021)
317. Liu, J., Musialski, P., Wonka, P., Ye, J.: Tensor completion for estimating missing values in visual data. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **35**(1), 208–220 (2012)
318. Liu, N., Wang, Z., Sun, M., Wang, H., Wang, B.: Numerical simulation of liquid droplet breakup in supersonic flows. *Acta Astronautica* **145**, 116–130 (2018)
319. Liu, P., Sun, S.: The application of artificial neural networks on the health monitoring of bridges. *Structural Health Monitoring, Current Status and Perspectives* pp. 103–110 (1997)

320. Liu, W.K., Gan, Z., Fleming, M.: Knowledge-driven dimension reduction and reduced order surrogate models. In: *Mechanistic Data Science for STEM Education and Applications*, pp. 131–170. Springer (2021)
321. Liu, X., Tian, S., Tao, F., Yu, W.: A review of artificial neural networks in the constitutive modeling of composite materials. *Composites Part B: Engineering* **224**, 109152 (2021)
322. Liu, Y., Kutz, J.N., Brunton, S.L.: Hierarchical deep learning of multiscale differential equation time-steppers. *arXiv preprint arXiv:2008.09768* (2020)
323. Liu, Y., Ponce, C., Brunton, S.L., Kutz, J.N.: Multiresolution convolutional autoencoders. *Journal of Computational Physics* p. 111801 (2022)
324. Logarzo, H.J., Capuano, G., Rimoli, J.J.: Smart constitutive laws: Inelastic homogenization through machine learning. *Computer Methods in Applied Mechanics and Engineering* **373**, 113482 (2021)
325. Lopez, E., Gonzalez, D., Aguado, J., Abisset-Chavanne, E., Cueto, E., Binetruy, C., Chinesta, F.: A manifold learning approach for integrated computational materials engineering. *Archives of Computational Methods in Engineering* **25**(1), 59–68 (2018)
326. Lore, K.G., Stoecklein, D., Davies, M., Ganapathysubramanian, B., Sarkar, S.: Hierarchical feature extraction for efficient design of microfluidic flow patterns. In: *Feature Extraction: Modern Questions and Challenges*, pp. 213–225. PMLR (2015)
327. Lorente, L., Vega, J., Velazquez, A.: Generation of aerodynamics databases using high-order singular value decomposition. *Journal of Aircraft* **45**(5), 1779–1788 (2008)
328. Lu, L., Jin, P., Pang, G., Zhang, Z., Karniadakis, G.E.: Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence* **3**(3), 218–229 (2021)
329. Lu, X., Liao, W., Zhang, Y., Huang, Y.: Intelligent structural design of shear wall residence using physics-enhanced generative adversarial networks. *Earthquake Engineering & Structural Dynamics* (2022)
330. Luo, H., Paal, S.G.: A locally weighted machine learning model for generalized prediction of drift capacity in seismic vulnerability assessments. *Computer-Aided Civil and Infrastructure Engineering* **34**(11), 935–950 (2019)
331. Luo, W., Hu, T., Ye, Y., Zhang, C., Wei, Y.: A hybrid predictive maintenance approach for CNC machine tool driven by digital twin. *Robotics and Computer-Integrated Manufacturing* **65**, 101974 (2020)
332. Lynch, M.E., Sarkar, S., Maute, K.: Machine learning to aid tuning of numerical parameters in topology optimization. *Journal of Mechanical Design* **141**(11) (2019)
333. Málaga-Chuquitaype, C.: Machine learning in structural design: an opinionated review. *Frontiers in Built Environment* p. 6 (2022)
334. Malik, M., Malik, M.K., Mehmood, K., Makhdoom, I.: Automatic speech recognition: a survey. *Multimedia Tools and Applications* **80**(6), 9411–9457 (2021)
335. Mallat, S.: Understanding deep convolutional networks. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **374**(2065), 20150203 (2016)
336. Manavalan, M.: Intersection of artificial intelligence, machine learning, and internet of things—an economic overview. *Global Disclosure of Economics and Business* **9**(2), 119–128 (2020)
337. Mangalathu, S., Jang, H., Hwang, S.H., Jeon, J.S.: Data-driven machine-learning-based seismic failure mode identification of reinforced concrete shear walls. *Engineering Structures* **208**, 110331 (2020)
338. Marr, B.: *Artificial intelligence in practice: how 50 successful companies used AI and machine learning to solve problems*. John Wiley & Sons (2019)
339. Martín, C.A., Méndez, A.C., Sainges, O., Petiot, E., Barasinski, A., Piana, M., Ratier, L., Chinesta, F.: Empowering design based on hybrid twin(TM): Application to acoustic resonators. *Designs* **4**(4), 44 (2020)
340. Massaroli, S., Poli, M., Califano, F., Faragasso, A., Park, J., Yamashita, A., Asama, H.: Port-hamiltonian approach to neural network training. In: *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 6799–6806. IEEE (2019)

341. Mattheakis, M., Sondak, D., Dogra, A.S., Protopapas, P.: Hamiltonian neural networks for solving equations of motion. *Physical Review E* **105**(6), 065305 (2022)
342. Maulik, R., Lusch, B., Balaprakash, P.: Reduced-order modeling of advection-dominated systems with recurrent neural networks and convolutional autoencoders. *Physics of Fluids* **33**(3), 037106 (2021)
343. Mayani, M.G., Svendsen, M., Oedegaard, S.: Drilling digital twin success stories the last 10 years. In: *SPE Norway One Day Seminar. OnePetro* (2018)
344. McCoy, L.G., Brenna, C.T., Chen, S.S., Vold, K., Das, S.: Believing in black boxes: Machine learning for healthcare does not need explainability to be evidence-based. *Journal of Clinical Epidemiology* **142**, 252–257 (2022)
345. McCulloch, W.S., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics* **5**(4), 115–133 (1943)
346. McInnes, L., Healy, J., Melville, J.: UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426* (2018)
347. Mehrjoo, M., Khaji, N., Moharrami, H., Bahreininejad, A.: Damage detection of truss bridge joints using artificial neural networks. *Expert Systems with Applications* **35**(3), 1122–1131 (2008)
348. Meijer, R.J., Goeman, J.J.: Efficient approximate k-fold and leave-one-out cross-validation for ridge regression. *Biometrical Journal* **55**(2), 141–155 (2013)
349. Meng, L., Breikopf, P., Quilliec, G.L., Raghavan, B., Villon, P.: Nonlinear shape-manifold learning approach: concepts, tools and applications. *Archives of Computational Methods in Engineering* **25**(1), 1–21 (2018)
350. Meng, L., McWilliams, B., Jarosinski, W., Park, H.Y., Jung, Y.G., Lee, J., Zhang, J.: Machine learning in additive manufacturing: a review. *JOM* **72**(6), 2363–2377 (2020)
351. Meng, X., Li, Z., Zhang, D., Karniadakis, G.E.: Ppinn: Parareal physics-informed neural network for time-dependent pdes. *Computer Methods in Applied Mechanics and Engineering* **370**, 113250 (2020)
352. Miñano, M., Montáns, F.J.: A new approach to modeling isotropic damage for Mullins effect in hyperelastic materials. *International Journal of Solids and Structures* **67**, 272–282 (2015)
353. Miao, P., Yokota, H.: Comparison of markov chain and recurrent neural network in predicting bridge deterioration considering various factors. *Structure and Infrastructure Engineering* pp. 1–13 (2022)
354. Miao, P., Yokota, H., Zhang, Y.: Deterioration prediction of existing concrete bridges using a lstm recurrent neural network. *Structure and Infrastructure Engineering* pp. 1–15 (2021)
355. Michalski, R.S., Carbonell, J.G., Mitchell, T.M.: Machine learning: An artificial intelligence approach. Springer Science & Business Media (2013)
356. Miñano, M., Montáns, F.J.: WYPiWYG damage mechanics for soft materials: A data-driven approach. *Archives of Computational Methods in Engineering* **25**(1), 165–193 (2018)
357. Mishra, M.: Machine learning techniques for structural health monitoring of heritage buildings: A state-of-the-art review and case studies. *Journal of Cultural Heritage* **47**, 227–245 (2021)
358. Mitchell, M.: An introduction to genetic algorithms. MIT press (1998)
359. Miyazawa, Y., Briffod, F., Shiraiwa, T., Enoki, M.: Prediction of cyclic stress–strain property of steels by crystal plasticity simulations and machine learning. *Materials* **12**(22), 3668 (2019)
360. Mohammadzadeh, S., Kim, Y., Ahn, J.: Pca-based neuro-fuzzy model for system identification of smart structures. *Journal of Smart Structures and Systems* **15**(5), 1139–1158 (2015)
361. Molnar, C., Casalicchio, G., Bischl, B.: iml: An R package for interpretable machine learning. *Journal of Open Source Software* **3**(26), 786 (2018)
362. Monostori, L., Márkus, A., Van Brussel, H., Westkämpfer, E.: Machine learning approaches to manufacturing. *CIRP annals* **45**(2), 675–712 (1996)
363. Morandin, R., Nicodemus, J., Unger, B.: Port-Hamiltonian dynamic mode decomposition. *arXiv preprint arXiv:2204.13474* (2022)
364. Moreno, S., Amores, V.J., Benítez, J.M., Montáns, F.J.: Reverse-engineering and modeling the 3d passive and active responses of skeletal muscle using a data-driven, non-parametric, spline-based procedure. *Journal of the Mechanical Behavior of Biomedical Materials* **110**, 103877 (2020)

365. Moroni, D., Pascali, M.A.: Learning topology: bridging computational topology and machine learning. *Pattern Recognition and Image Analysis* **31**(3), 443–453 (2021)
366. Moya, B., Alfaro, I., Gonzalez, D., Chinesta, F., Cueto, E.: Physically sound, self-learning digital twins for sloshing fluids. *Plos One* **15**(6), e0234569 (2020)
367. Moya, B., Badías, A., Alfaro, I., Chinesta, F., Cueto, E.: Digital twins that learn and correct themselves. *International Journal for Numerical Methods in Engineering* **123**(13), 3034–3044 (2022)
368. Mozaffar, M., Bostanabad, R., Chen, W., Ehmann, K., Cao, J., Bessa, M.: Deep learning predicts path-dependent plasticity. *Proceedings of the National Academy of Sciences* **116**(52), 26414–26420 (2019)
369. Mukherjee, S., Lu, D., Raghavan, B., Breitkopf, P., Dutta, S., Xiao, M., Zhang, W.: Accelerating large-scale topology optimization: state-of-the-art and challenges. *Archives of Computational Methods in Engineering* **28**(7), 4549–4571 (2021)
370. Muñoz, D., Nadal, E., Albelda, J., Chinesta, F., Ródenas, J.: Allying topology and shape optimization through machine learning algorithms. *Finite Elements in Analysis and Design* **204**, 103719 (2022)
371. Murata, T., Fukami, K., Fukagata, K.: Nonlinear mode decomposition with convolutional neural networks for fluid dynamics. *Journal of Fluid Mechanics* **882** (2020)
372. Muthali, A., Laine, F., Tomlin, C.: Incorporating data uncertainty in object tracking algorithms. *arXiv preprint arXiv:2109.10521* (2021)
373. Nascimento, R.G., Viana, F.A.: Cumulative damage modeling with recurrent neural networks. *AIAA Journal* **58**(12), 5459–5471 (2020)
374. Nasiri, S., Khosravani, M.R., Weinberg, K.: Fracture mechanics and mechanical fault detection by artificial intelligence methods: A review. *Engineering Failure Analysis* **81**, 270–293 (2017)
375. Nassif, A.B., Talib, M.A., Nassir, Q., Albadani, H., Albab, F.D.: Machine learning for cloud security: A systematic review. *IEEE Access* (2021)
376. Nawafleh, N., AL-Oqla, F.M.: Artificial neural network for predicting the mechanical performance of additive manufacturing thermoset carbon fiber composite materials. *Journal of the Mechanical Behavior of Materials* **31**(1), 501–513 (2022)
377. Nayak, H.D., Anvitha, L., Shetty, A., D'Souza, D.J., Abraham, M.P., et al.: Fraud detection in online transactions using machine learning approaches—a review. *Advances in Artificial Intelligence and Data Engineering* pp. 589–599 (2021)
378. Nayak, S., Lyngdoh, G.A., Shukla, A., Das, S.: Predicting the near field underwater explosion response of coated composite cylinders using multiscale simulations, experiments, and machine learning. *Composite Structures* **283**, 115157 (2022)
379. Nguyen, D.H., Nguyen, Q.B., Bui-Tien, T., De Roeck, G., Wahab, M.A.: Damage detection in girder bridges using modal curvatures gapped smoothing method and convolutional neural network: Application to bo nghi bridge. *Theoretical and Applied Fracture Mechanics* **109**, 102728 (2020)
380. Nguyen, L.T.K., Keip, M.A.: A data-driven approach to nonlinear elasticity. *Computers & Structures* **194**, 97–115 (2018)
381. Nguyen-Thanh, V.M., Zhuang, X., Rabczuk, T.: A deep energy method for finite deformation hyperelasticity. *European Journal of Mechanics-A/Solids* **80**, 103874 (2020)
382. Ni, F., Zhang, J., Noori, M.N.: Deep learning for data anomaly detection and data compression of a long-span suspension bridge. *Computer-Aided Civil and Infrastructure Engineering* **35**(7), 685–700 (2020)
383. Ni, Y.Q., Jiang, S., Ko, J.M.: 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, pp. 347–356. SPIE (2001)
384. Nick, H., Aziminejad, A., Hosseini, M.H., Laknejadi, K.: Damage identification in steel girder bridges using modal strain energy-based damage index method and artificial neural network. *Engineering Failure Analysis* **119**, 105010 (2021)
385. Oh, S., Jung, Y., Kim, S., Lee, I., Kang, N.: Deep generative design: Integration of topology optimization and generative models. *Journal of Mechanical Design* **141**(11) (2019)

386. Olivier, A., Shields, M.D., Graham-Brady, L.: Bayesian neural networks for uncertainty quantification in data-driven materials modeling. *Computer Methods in Applied Mechanics and Engineering* **386**, 114079 (2021)
387. Omairi, A., Ismail, Z.H.: Towards machine learning for error compensation in additive manufacturing. *Applied Sciences* **11**(5), 2375 (2021)
388. Ongsulee, P.: Artificial intelligence, machine learning and deep learning. In: 2017 15th International Conference on ICT and Knowledge Engineering (ICT&KE), pp. 1–6. IEEE (2017)
389. Paluszek, M., Thomas, S.: MATLAB machine learning. Apress (2016)
390. Panagiotopoulos, P., Waszczyszyn, Z.: The neural network approach in plasticity and fracture mechanics. In: *Neural Networks in the Analysis and Design of Structures*, pp. 161–195. Springer (1999)
391. Panakkat, A., Adeli, H.: Recurrent neural network for approximate earthquake time and location prediction using multiple seismicity indicators. *Computer-Aided Civil and Infrastructure Engineering* **24**(4), 280–292 (2009)
392. Pang, G., Lu, L., Karniadakis, G.E.: fPINNs: Fractional physics-informed neural networks. *SIAM Journal on Scientific Computing* **41**(4), A2603–A2626 (2019)
393. Paszkowicz, W.: Genetic algorithms, a nature-inspired tool: survey of applications in materials science and related fields. *Materials and Manufacturing Processes* **24**(2), 174–197 (2009)
394. Pathak, J., Subramanian, S., Harrington, P., Raja, S., Chattopadhyay, A., Mardani, M., Kurth, T., Hall, D., Li, Z., Azizzadenesheli, K., et al.: Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators. *arXiv preprint arXiv:2202.11214* (2022)
395. Pathan, M., Ponnusami, S., Pathan, J., Pitisongsawat, R., Erice, B., Petrinic, N., Tagarielli, V.: Predictions of the mechanical properties of unidirectional fibre composites by supervised machine learning. *Scientific Reports* **9**(1), 1–10 (2019)
396. Pawar, S., San, O., Nair, A., Rasheed, A., Kvamsdal, T.: Model fusion with physics-guided machine learning: Projection-based reduced-order modeling. *Physics of Fluids* **33**(6), 067123 (2021)
397. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine Learning in Python. *The Journal of Machine Learning Research* **12**, 2825–2830 (2011)
398. Peng, G.C., Alber, M., Buganza Tepole, A., Cannon, W.R., De, S., Dura-Bernal, S., Garikipati, K., Karniadakis, G., Lytton, W.W., Perdikaris, P., et al.: Multiscale modeling meets machine learning: What can we learn? *Archives of Computational Methods in Engineering* **28**(3), 1017–1037 (2021)
399. Penumuru, D.P., Muthuswamy, S., Karumbu, P.: Identification and classification of materials using machine vision and machine learning in the context of industry 4.0. *Journal of Intelligent Manufacturing* **31**(5), 1229–1241 (2020)
400. Pereira, D.R., Piteri, M.A., Souza, A.N., Papa, J.P., Adeli, H.: Fema: A finite element machine for fast learning. *Neural Computing and Applications* **32**(10), 6393–6404 (2020)
401. Pham, D.T., Afify, A.A.: Machine-learning techniques and their applications in manufacturing. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* **219**(5), 395–412 (2005)
402. PK, F.A.: What is artificial intelligence? “Success is no accident. It is hard work, perseverance, learning, studying, sacrifice and most of all, love of what you are doing or learning to do”. p. 65 (1984)
403. Platzer, A., Leygue, A., Stainier, L., Ortiz, M.: Finite element solver for data-driven finite strain elasticity. *Computer Methods in Applied Mechanics and Engineering* **379**, 113756 (2021)
404. Proctor, J.L., Brunton, S.L., Kutz, J.N.: Dynamic mode decomposition with control. *SIAM Journal on Applied Dynamical Systems* **15**(1), 142–161 (2016)
405. Qin, J., Hu, F., Liu, Y., Witherell, P., Wang, C.C., Rosen, D.W., Simpson, T., Lu, Y., Tang, Q.: Research and application of machine learning for additive manufacturing. *Additive Manufacturing* p. 102691 (2022)

406. Quqa, S., Martakis, P., Movsessian, A., Pai, S., Reuland, Y., Chatzi, E.: Two-step approach for fatigue crack detection in steel bridges using convolutional neural networks. *Journal of Civil Structural Health Monitoring* **12**(1), 127–140 (2022)
407. Rabin, N., Fishelov, D.: Missing data completion using diffusion maps and Laplacian pyramids. In: *International Conference on Computational Science and Its Applications*, pp. 284–297. Springer (2017)
408. Rai, R., Sahu, C.K.: 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 (2020)
409. Rai, R., Tiwari, M.K., Ivanov, D., Dolgui, A.: *Machine learning in manufacturing and industry 4.0 applications* (2021)
410. Raissi, M., Karniadakis, G.E.: Hidden physics models: Machine learning of nonlinear partial differential equations. *Journal of Computational Physics* **357**, 125–141 (2018)
411. Raissi, M., Perdikaris, P., Karniadakis, G.E.: Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics* **378**, 686–707 (2019)
412. Ramoni, M., Sebastiani, P.: Robust learning with missing data. *Machine Learning* **45**(2), 147–170 (2001)
413. Randle, D., Protopapas, P., Sondak, D.: Unsupervised learning of solutions to differential equations with generative adversarial networks. *arXiv preprint arXiv:2007.11133* (2020)
414. Ranković, V., Grujović, N., Divac, D., Milivojević, N., Novaković, A.: Modelling of dam behaviour based on neuro-fuzzy identification. *Engineering Structures* **35**, 107–113 (2012)
415. Rao, C., Sun, H., Liu, Y.: Physics-informed deep learning for computational elastodynamics without labeled data. *Journal of Engineering Mechanics* **147**(8), 04021043 (2021)
416. Raschka, S.: *Python machine learning*. Packt publishing ltd (2015)
417. Raschka, S., Mirjalili, V.: *Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2*. Packt Publishing Ltd (2019)
418. Razvi, S.S., Feng, S., Narayanan, A., Lee, Y.T.T., Witherell, P.: 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 (2019)
419. Reagan, D., Sabato, A., Niezrecki, C.: Feasibility of using digital image correlation for unmanned aerial vehicle structural health monitoring of bridges. *Structural Health Monitoring* **17**(5), 1056–1072 (2018)
420. Regan, T., Canturk, R., Slavkovsky, E., Niezrecki, C., Inalpolat, M.: 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 (2016)
421. Regazzoni, F., Dedè, L., Quarteroni, A.: Machine learning of multiscale active force generation models for the efficient simulation of cardiac electromechanics. *Computer Methods in Applied Mechanics and Engineering* **370**, 113268 (2020)
422. Ren, T., Wang, L., Chang, C., Li, X.: Machine learning-assisted multiphysics coupling performance optimization in a photocatalytic hydrogen production system. *Energy Conversion and Management* **216**, 112935 (2020)
423. Rice, L., Wong, E., Kolter, Z.: Overfitting in adversarially robust deep learning. In: *International Conference on Machine Learning*, pp. 8093–8104. PMLR (2020)
424. Robert, C.: *Machine learning, a probabilistic perspective* (2014)
425. Rocha, I., Kerfriden, P., van der Meer, F.: Micromechanics-based surrogate models for the response of composites: A critical comparison between a classical mesoscale constitutive model, hyper-reduction and neural networks. *European Journal of Mechanics-A/Solids* **82**, 103995 (2020)
426. Rocha, I., Kerfriden, P., van der Meer, F.: On-the-fly construction of surrogate constitutive models for concurrent multiscale mechanical analysis through probabilistic machine learning. *Journal of Computational Physics: X* **9**, 100083 (2021)

427. Rodríguez, M., Kramer, T.: Machine learning of two-dimensional spectroscopic data. *Chemical Physics* **520**, 52–60 (2019)
428. Rogers, T., Holmes, G., Cross, E., Worden, K.: On a grey box modelling framework for nonlinear system identification. In: *Special Topics in Structural Dynamics*, Volume 6, pp. 167–178. Springer (2017)
429. Roisman, I., Breitenbach, J., Tropea, C.: Thermal atomisation of a liquid drop after impact onto a hot substrate. *Journal of Fluid Mechanics* **842**, 87–101 (2018)
430. Romero, X., Latorre, M., Montáns, F.J.: Determination of the WYPiWYG strain energy density of skin through finite element analysis of the experiments on circular specimens. *Finite Elements in Analysis and Design* **134**, 1–15 (2017)
431. Rosafalco, L., Torzoni, M., Manzoni, A., Mariani, S., Corigliano, A.: Online structural health monitoring by model order reduction and deep learning algorithms. *Computers & Structures* **255**, 106604 (2021)
432. Rosenblatt, F.: The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review* **65**(6), 386 (1958)
433. Rosti, A., Rota, M., Penna, A.: An empirical seismic vulnerability model. *Bulletin of Earthquake Engineering* pp. 1–27 (2022)
434. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. *science* **290**(5500), 2323–2326 (2000)
435. Rowley, C.W.: Model reduction for fluids, using balanced proper orthogonal decomposition. *International Journal of Bifurcation and Chaos* **15**(03), 997–1013 (2005)
436. Rubio, P.B., Chamoin, L., Louf, F.: Real-time data assimilation and control on mechanical systems under uncertainties. *Advanced Modeling and Simulation in Engineering Sciences* **8**(1), 1–25 (2021)
437. Rudy, S.H., Brunton, S.L., Proctor, J.L., Kutz, J.N.: Data-driven discovery of partial differential equations. *Science Advances* **3**(4), e1602614 (2017)
438. Ruggieri, S., Cardellicchio, A., Leggieri, V., Uva, G.: Machine-learning based vulnerability analysis of existing buildings. *Automation in Construction* **132**, 103936 (2021)
439. Salazar, F., Morán, R., Toledo, M.Á., Oñate, E.: Data-based models for the prediction of dam behaviour: a review and some methodological considerations. *Archives of Computational Methods in Engineering* **24**(1), 1–21 (2017)
440. Salazar, F., Toledo, M., Oñate, E., Morán, R.: An empirical comparison of machine learning techniques for dam behaviour modelling. *Structural Safety* **56**, 9–17 (2015)
441. Salazar, F., Toledo, M.Á., Oñate, E., Suárez, B.: Interpretation of dam deformation and leakage with boosted regression trees. *Engineering Structures* **119**, 230–251 (2016)
442. Salloum, S.A., Alshurideh, M., Elnagar, A., Shaalan, K.: Machine learning and deep learning techniques for cybersecurity: a review. In: *The International Conference on Artificial Intelligence and Computer Vision*, pp. 50–57. Springer (2020)
443. Salman, O., Elhajj, I.H., Chehab, A., Kayssi, A.: A machine learning based framework for IoT device identification and abnormal traffic detection. *Transactions on Emerging Telecommunications Technologies* **33**(3), e3743 (2022)
444. Salman, O., Elhajj, I.H., Kayssi, A., Chehab, A.: A review on machine learning-based approaches for internet traffic classification. *Annals of Telecommunications* **75**(11), 673–710 (2020)
445. Sancarlos, A., Cameron, M., Le Peuvedic, J.M., Groulier, J., Duval, J.L., Cueto, E., Chinesta, F.: Learning stable reduced-order models for hybrid twins. *Data-Centric Engineering* **2** (2021)
446. Sankarasrinivasan, S., Balasubramanian, E., Karthik, K., Chandrasekar, U., Gupta, R.: Health monitoring of civil structures with integrated uav and image processing system. *Procedia Computer Science* **54**, 508–515 (2015)
447. Sarmadi, H., Karamodin, A.: A novel anomaly detection method based on adaptive mahalanobis-squared distance and one-class knn rule for structural health monitoring under environmental effects. *Mechanical Systems and Signal Processing* **140**, 106495 (2020)
448. Scardovelli, R., Zaleski, S.: Direct numerical simulation of free-surface and interfacial flow. *Annual Review of Fluid Mechanics* **31**(1), 567–603 (1999)

449. Schmid, P.J.: Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics* **656**, 5–28 (2010)
450. Schmid, P.J.: Application of the dynamic mode decomposition to experimental data. *Experiments in Fluids* **50**(4), 1123–1130 (2011)
451. Schmid, P.J., Li, L., Juniper, M.P., Pust, O.: Applications of the dynamic mode decomposition. *Theoretical and Computational Fluid Dynamics* **25**(1), 249–259 (2011)
452. Schmidt, M., Lipson, H.: Distilling free-form natural laws from experimental data. *Science* **324**(5923), 81–85 (2009)
453. Schmidt, M., Lipson, H.: Symbolic regression of implicit equations. In: *Genetic Programming Theory and Practice VII*, pp. 73–85. Springer (2010)
454. Schölkopf, B., Herbrich, R., Smola, A.J.: A generalized representer theorem. In: *International conference on computational learning theory*, pp. 416–426. Springer (2001)
455. Schölkopf, B., Smola, A., Müller, K.R.: Kernel principal component analysis. In: *International Conference on Artificial Neural Networks*, pp. 583–588. Springer (1997)
456. Searson, D.: GPTIPS: Genetic programming and symbolic regression for matlab. <https://sites.google.com/site/gptips4matlab/?pli=1> (2009)
457. Seff, A., Zhou, W., Richardson, N., Adams, R.P.: Vitruvion: A generative model of parametric cad sketches. *arXiv preprint arXiv:2109.14124* (2021)
458. Seibi, A., Al-Alawi, S.: Prediction of fracture toughness using artificial neural networks (anns). *Engineering Fracture Mechanics* **56**(3), 311–319 (1997)
459. Sevieri, G., De Falco, A.: Dynamic structural health monitoring for concrete gravity dams based on the bayesian inference. *Journal of Civil Structural Health Monitoring* **10**(2), 235–250 (2020)
460. Sharma, S., Bhatt, M., Sharma, P.: Face recognition system using machine learning algorithm. In: *2020 5th International Conference on Communication and Electronics Systems (ICCES)*, pp. 1162–1168. IEEE (2020)
461. Sharp, M., Ak, R., Hedberg Jr, T.: A survey of the advancing use and development of machine learning in smart manufacturing. *Journal of Manufacturing Systems* **48**, 170–179 (2018)
462. Shihavuddin, A., Chen, X., Fedorov, V., Nymark Christensen, A., Andre Brogaard Riis, N., Branner, K., BJORHOLM DAHL, A., Reinhold Paulsen, R.: Wind turbine surface damage detection by deep learning aided drone inspection analysis. *Energies* **12**(4), 676 (2019)
463. Shin, D., Yoo, S., Lee, S., Kim, M., Hwang, K.H., Park, J.H., Kang, N.: 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 (2021)
464. Shorten, C., Khoshgoftaar, T.M.: A survey on image data augmentation for deep learning. *Journal of Big Data* **6**(1), 1–48 (2019)
465. Shu, D., Cunningham, J., Stump, G., Miller, S.W., Yukish, M.A., Simpson, T.W., Tucker, C.S.: 3D design using generative adversarial networks and physics-based validation. *Journal of Mechanical Design* **142**(7), 071701 (2020)
466. Sigmund, O.: Systematic design of metamaterials by topology optimization. In: *IUTAM symposium on modelling nanomaterials and nanosystems*, pp. 151–159. Springer (2009)
467. Silva, M., Santos, A., Figueiredo, E., Santos, R., Sales, C., Costa, J.C.: A novel unsupervised approach based on a genetic algorithm for structural damage detection in bridges. *Engineering Applications of Artificial Intelligence* **52**, 168–180 (2016)
468. Simpson, T., Dervilis, N., Chatzi, E.: Machine learning approach to model order reduction of nonlinear systems via autoencoder and LSTM networks. *Journal of Engineering Mechanics* **147**(10), 04021061 (2021)
469. Singh, A.P., Medida, S., Duraisamy, K.: Machine-learning-augmented predictive modeling of turbulent separated flows over airfoils. *AIAA journal* **55**(7), 2215–2227 (2017)
470. Singh, H., Gupta, M., Mahajan, P.: Reduced order multiscale modeling of fiber reinforced polymer composites including plasticity and damage. *Mechanics of Materials* **111**, 35–56 (2017)
471. Sirca Jr, G., Adeli, H.: System identification in structural engineering. *Scientia Iranica* **19**(6), 1355–1364 (2012)

472. Soize, C., Ghanem, R.: Physics-constrained non-Gaussian probabilistic learning on manifolds. *International Journal for Numerical Methods in Engineering* **121**(1), 110–145 (2020)
473. Sordoni, A., Bengio, Y., Vahabi, H., Lioma, C., Grue Simonsen, J., Nie, J.Y.: 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 (2015)
474. Sorini, A., Pineda, E.J., Stuckner, J., Gustafson, P.A.: A convolutional neural network for multiscale modeling of composite materials. In: *AIAA Scitech 2021 Forum*, p. 0310 (2021)
475. Spalart, P., Allmaras, S.: A one-equation turbulence model for aerodynamic flows. In: *30th aerospace sciences meeting and exhibit*, p. 439 (1992)
476. Speziale, C.G.: Turbulence modeling for time-dependent RANS and VLES: a review. *AIAA Journal* **36**(2), 173–184 (1998)
477. Sprangers, O., Babuška, R., Nagesh Rao, S.P., Lopes, G.A.: Reinforcement learning for port-Hamiltonian systems. *IEEE Transactions on Cybernetics* **45**(5), 1017–1027 (2014)
478. Stahl, B.C.: *Artificial Intelligence for a Better Future: An Ecosystem Perspective on the Ethics of AI and Emerging Digital Technologies*. Springer Nature (2021)
479. Stainier, L., Leygue, A., Ortiz, M.: Model-free data-driven methods in mechanics: material data identification and solvers. *Computational Mechanics* **64**(2), 381–393 (2019)
480. 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)
481. Stevens, B., Colonius, T.: Enhancement of shock-capturing methods via machine learning. *Theoretical and Computational Fluid Dynamics* **34**(4), 483–496 (2020)
482. Stoll, A., Benner, P.: Machine learning for material characterization with an application for predicting mechanical properties. *GAMM-Mitteilungen* **44**(1), e202100003 (2021)
483. Straus, J., Skogestad, S.: Variable reduction for surrogate modelling. *Proceedings of the Foundations of Computer-Aided Process Operations*, Tucson, AZ, USA pp. 8–12 (2017)
484. Ströfer, C.M., Wu, J., Xiao, H., Paterson, E.: Data-driven, physics-based feature extraction from fluid flow fields using convolutional neural networks. *Communications in Computational Physics* **25**(3), 625–650 (2018)
485. Sun, F., Liu, Y., Sun, H.: Physics-informed spline learning for nonlinear dynamics discovery. *arXiv preprint arXiv:2105.02368* (2021)
486. Sun, H., Burton, H.V., Huang, H.: Machine learning applications for building structural design and performance assessment: State-of-the-art review. *Journal of Building Engineering* **33**, 101816 (2021)
487. Surjadi, J.U., Gao, L., Du, H., Li, X., Xiong, X., Fang, N.X., Lu, Y.: Mechanical metamaterials and their engineering applications. *Advanced Engineering Materials* **21**(3), 1800864 (2019)
488. Sussman, T., Bathe, K.J.: A model of incompressible isotropic hyperelastic material behavior using spline interpolations of tension–compression test data. *Communications in Numerical Methods in Engineering* **25**(1), 53–63 (2009)
489. Swischuk, R., Allaire, D.: A machine learning approach to aircraft sensor error detection and correction. *Journal of Computing and Information Science in Engineering* **19**(4) (2019)
490. Tam, K.M.M., Moosavi, V., Van Mele, T., Block, P.: 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) (2020)
491. Tamke, M., Nicholas, P., Zwierzycki, M.: Machine learning for architectural design: Practices and infrastructure. *International Journal of Architectural Computing* **16**(2), 123–143 (2018)
492. Tang, H.S., Xue, S.T., Chen, R., Sato, T.: Online weighted LS-SVM for hysteretic structural system identification. *Engineering Structures* **28**(12), 1728–1735 (2006)
493. Tang, Q., Dang, J., Cui, Y., Wang, X., Jia, J.: Machine learning-based fast seismic risk assessment of building structures. *Journal of Earthquake Engineering* pp. 1–22 (2021)
494. Tartakovsky, A.M., Marrero, C.O., Perdikaris, P., Tartakovsky, G.D., Barajas-Solano, D.: Learning parameters and constitutive relationships with physics informed deep neural networks. *arXiv preprint arXiv:1808.03398* (2018)

- 495. Tharwat, A.: Linear vs. quadratic discriminant analysis classifier: a tutorial. *International Journal of Applied Pattern Recognition* **3**(2), 145–180 (2016)
- 496. Theocaris, P., Panagiotopoulos, P.: Neural networks for computing in fracture mechanics. methods and prospects of applications. *Computer Methods in Applied Mechanics and Engineering* **106**(1-2), 213–228 (1993)
- 497. Ti, Z., Deng, X.W., Yang, H.: Wake modeling of wind turbines using machine learning. *Applied Energy* **257**, 114025 (2020)
- 498. Tian, C., Li, T., Bustillos, J., Bhattacharya, S., Turnham, T., Yeo, J., Moridi, A.: Data-driven approaches toward smarter additive manufacturing. *Advanced Intelligent Systems* **3**(12), 2100014 (2021)
- 499. Tibshirani, R.: Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society: Series B (Methodological)* **58**(1), 267–288 (1996)
- 500. Tikhonov, A.N.: 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 (1963)
- 501. Torky, A.A., Ohno, S.: Deep learning techniques for predicting nonlinear multi-component seismic responses of structural buildings. *Computers & Structures* **252**, 106570 (2021)
- 502. Trincherio, R., Larbi, M., Torun, H.M., Canavero, F.G., Swaminathan, M.: Machine learning and uncertainty quantification for surrogate models of integrated devices with a large number of parameters. *IEEE Access* **7**, 4056–4066 (2018)
- 503. Tsur, E.E.: Computer-aided design of microfluidic circuits. *Annual Review Biomedical Engineering* **22**, 285–307 (2020)
- 504. Tu, J.H.: Dynamic mode decomposition: Theory and applications. Ph.D. thesis, Princeton University (2013)
- 505. Turaga, P., Anirudh, R., Chellappa, R.: Manifold learning. *Computer Vision: A Reference Guide* pp. 1–6 (2020)
- 506. Tzonis, A., White, I.: *Automation Based Creative Design-Research and Perspectives*. Newnes (2012)
- 507. Vafaie, H., De Jong, K.A.: Genetic algorithms as a tool for feature selection in machine learning. In: *ICTAI*, pp. 200–203 (1992)
- 508. Valdés-Alonzo, G., Binetruy, C., Eck, B., García-González, A., Leygue, A.: Phase distribution and properties identification of heterogeneous materials: A data-driven approach. *Computer Methods in Applied Mechanics and Engineering* **390**, 114354 (2022)
- 509. Van Der Schaft, A., Jeltsema, D., et al.: Port-Hamiltonian systems theory: An introductory overview. *Foundations and Trends® in Systems and Control* **1**(2-3), 173–378 (2014)
- 510. Van Erven, T., Harremos, P.: Rényi divergence and Kullback-Leibler divergence. *IEEE Transactions on Information Theory* **60**(7), 3797–3820 (2014)
- 511. Vassallo, D., Krishnamurthy, R., Fernando, H.J.: Utilizing physics-based input features within a machine learning model to predict wind speed forecasting error. *Wind Energy Science* **6**(1), 295–309 (2021)
- 512. Verkhivker, G.M., Agajanian, S., Hu, G., Tao, P.: Allosteric regulation at the crossroads of new technologies: multiscale modeling, networks, and machine learning. *Frontiers in Molecular Biosciences* **7**, 136 (2020)
- 513. Vitola, J., Pozo, F., Tibaduiza, D.A., Anaya, M.: A sensor data fusion system based on k-nearest neighbor pattern classification for structural health monitoring applications. *Sensors* **17**(2), 417 (2017)
- 514. Vivanco-Benavides, L.E., Martínez-González, C.L., Mercado-Zúñiga, C., Torres-Torres, C.: Machine learning and materials informatics approaches in the analysis of physical properties of carbon nanotubes: a review. *Computational Materials Science* **201**, 110939 (2022)
- 515. Vlassis, N.N., Ma, R., Sun, W.: Geometric deep learning for computational mechanics part i: Anisotropic hyperelasticity. *Computer Methods in Applied Mechanics and Engineering* **371**, 113299 (2020)
- 516. Vlassis, N.N., Sun, W.: Sobolev training of thermodynamic-informed neural networks for interpretable elasto-plasticity models with level set hardening. *Computer Methods in Applied Mechanics and Engineering* **377**, 113695 (2021)

517. Volpiani, P.S., Meyer, M., Franceschini, L., Dandois, J., Renac, F., Martin, E., Marquet, O., Sipp, D.: Machine learning-augmented turbulence modeling for RANS simulations of massively separated flows. *Physical Review Fluids* **6**(6), 064607 (2021)
518. Wackers, J., Visonneau, M., Serani, A., Pellegrini, R., Broglia, R., Diez, M.: Multi-fidelity machine learning from adaptive-and multi-grid rans simulations. In: 33rd Symposium on Naval Hydrodynamics (2020)
519. Wang, C., Tan, X., Tor, S., Lim, C.: Machine learning in additive manufacturing: State-of-the-art and perspectives. *Additive Manufacturing* **36**, 101538 (2020)
520. Wang, C., Xiao, J., Zhang, C., Xiao, X.: Structural health monitoring and performance analysis of a 12-story recycled aggregate concrete structure. *Engineering Structures* **205**, 110102 (2020)
521. Wang, C., Xu, L.y., Fan, J.s.: A general deep learning framework for history-dependent response prediction based on UA-Seq2Seq model. *Computer Methods in Applied Mechanics and Engineering* **372**, 113357 (2020)
522. Wang, H., O'Brien, J.F., Ramamoorthi, R.: Data-driven elastic models for cloth: modeling and measurement. *ACM Transactions on Graphics (TOG)* **30**(4), 1–12 (2011)
523. Wang, J.X., Wu, J., Ling, J., Iaccarino, G., Xiao, H.: A comprehensive physics-informed machine learning framework for predictive turbulence modeling. *arXiv preprint arXiv:1701.07102* (2017)
524. Wang, K., Sun, W.: A multiscale multi-permeability poroplasticity model linked by recursive homogenizations and deep learning. *Computer Methods in Applied Mechanics and Engineering* **334**, 337–380 (2018)
525. Wang, L., Zhang, Z., Long, H., Xu, J., Liu, R.: Wind turbine gearbox failure identification with deep neural networks. *IEEE Transactions on Industrial Informatics* **13**(3), 1360–1368 (2016)
526. Wang, L., et al.: Application and development prospect of digital twin technology in aerospace. *IFAC-PapersOnLine* **53**(5), 732–737 (2020)
527. Wang, Q., Guo, Y., Yu, L., Li, P.: Earthquake prediction based on spatio-temporal data mining: an lstm network approach. *IEEE Transactions on Emerging Topics in Computing* **8**(1), 148–158 (2017)
528. Wang, Y., Cheung, S.W., Chung, E.T., Efendiev, Y., Wang, M.: Deep multiscale model learning. *Journal of Computational Physics* **406**, 109071 (2020)
529. Wang, Y., Ghaboussi, J., Hoerig, C., Insana, M.F.: A data-driven approach to characterizing nonlinear elastic behavior of soft materials. *Journal of the Mechanical Behavior of Biomedical Materials* **130**, 105178 (2022)
530. Waszczyszyn, Z., Ziemiański, L.: Neural networks in mechanics of structures and materials—new results and prospects of applications. *Computers & Structures* **79**(22–25), 2261–2276 (2001)
531. Weiss, J.A., Maker, B.N., Govindjee, S.: Finite element implementation of incompressible, transversely isotropic hyperelasticity. *Computer Methods in Applied Mechanics and Engineering* **135**(1–2), 107–128 (1996)
532. White, D.A., Arrighi, W.J., Kudo, J., Watts, S.E.: Multiscale topology optimization using neural network surrogate models. *Computer Methods in Applied Mechanics and Engineering* **346**, 1118–1135 (2019)
533. Widrow, B., Hoff, M.E.: Associative storage and retrieval of digital information in networks of adaptive “neurons”. In: *Biological Prototypes and Synthetic Systems*, pp. 160–160. Springer (1962)
534. Williams, M.O., Kevrekidis, I.G., Rowley, C.W.: A data-driven approximation of the koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science* **25**(6), 1307–1346 (2015)
535. Wilt, J.K., Yang, C., Gu, G.X.: Accelerating auxetic metamaterial design with deep learning. *Advanced Engineering Materials* **22**(5), 1901266 (2020)
536. Wirtz, D., Karajan, N., Haasdonk, B.: Surrogate modeling of multiscale models using kernel methods. *International Journal for Numerical Methods in Engineering* **101**(1), 1–28 (2015)

537. Wood, M.A., Cusentino, M.A., Wirth, B.D., Thompson, A.P.: Data-driven material models for atomistic simulation. *Physical Review B* **99**(18), 184305 (2019)
538. Wu, D., Wei, Y., Terpenney, J.: 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 (2018)
539. Wu, J.L., Xiao, H., Paterson, E.: Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework. *Physical Review Fluids* **3**(7), 074602 (2018)
540. Wu, L., Liu, L., Wang, Y., Zhai, Z., Zhuang, H., Krishnaraju, D., Wang, Q., Jiang, H.: A machine learning-based method to design modular metamaterials. *Extreme Mechanics Letters* **36**, 100657 (2020)
541. Wu, L., Zulueta, K., Major, Z., Arriaga, A., Noels, L.: Bayesian inference of non-linear multiscale model parameters accelerated by a deep neural network. *Computer Methods in Applied Mechanics and Engineering* **360**, 112693 (2020)
542. Wu, R.T., Jahanshahi, M.R.: Data fusion approaches for structural health monitoring and system identification: past, present, and future. *Structural Health Monitoring* **19**(2), 552–586 (2020)
543. Wu, X., Park, Y., Li, A., Huang, X., Xiao, F., Usmani, A.: Smart detection of fire source in tunnel based on the numerical database and artificial intelligence. *Fire Technology* **57**(2), 657–682 (2021)
544. Wu, Y., Sui, Y., Wang, G.: Vision-based real-time aerial object localization and tracking for uav sensing system. *IEEE Access* **5**, 23969–23978 (2017)
545. Xames, M.D., Torsha, F.K., Sarwar, F.: A systematic literature review on recent trends of machine learning applications in additive manufacturing. *Journal of Intelligent Manufacturing* pp. 1–27 (2022)
546. Xiao, S., Hu, R., Li, Z., Attarian, S., Björk, K.M., Lendasse, A.: A machine-learning-enhanced hierarchical multiscale method for bridging from molecular dynamics to continua. *Neural Computing and Applications* **32**(18), 14359–14373 (2020)
547. Xie, T., Grossman, J.C.: Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. *Physical Review Letters* **120**(14), 145301 (2018)
548. Xie, X., Bennett, J., Saha, S., Lu, Y., Cao, J., Liu, W.K., Gan, Z.: Mechanistic data-driven prediction of as-built mechanical properties in metal additive manufacturing. *NPJ Computational Materials* **7**(1), 1–12 (2021)
549. Xie, Y., Ebad Sichani, M., Padgett, J.E., DesRoches, R.: The promise of implementing machine learning in earthquake engineering: A state-of-the-art review. *Earthquake Spectra* **36**(4), 1769–1801 (2020)
550. Xiong, W., Wu, L., Alleva, F., Droppo, J., Huang, X., Stolcke, A.: The Microsoft 2017 conversational speech recognition system. In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5934–5938. IEEE (2018)
551. Xu, C., Cao, B.T., Yuan, Y., Meschke, G.: Transfer learning based physics-informed neural networks for solving inverse problems in tunneling. *arXiv preprint arXiv:2205.07731* (2022)
552. Xu, H., Caramanis, C., Mannor, S.: Robust regression and Lasso. *Advances in Neural Information Processing Systems* **21** (2008)
553. Xu, J., Duraisamy, K.: Multi-level convolutional autoencoder networks for parametric prediction of spatio-temporal dynamics. *Computer Methods in Applied Mechanics and Engineering* **372**, 113379 (2020)
554. Yadav, D., Salmani, S.: Deepfake: A survey on facial forgery technique using generative adversarial network. In: *2019 International conference on intelligent computing and control systems (ICCS)*, pp. 852–857. IEEE (2019)
555. Yagawa, G., Okuda, H.: Neural networks in computational mechanics. *Archives of Computational Methods in Engineering* **3**(4), 435–512 (1996)
556. Yamaguchi, T., Okuda, H.: Zooming method for fea using a neural network. *Computers & Structures* **247**, 106480 (2021)

557. Yan, C., Vescovini, R., Dozio, L.: A framework based on physics-informed neural networks and extreme learning for the analysis of composite structures. *Computers & Structures* **265**, 106761 (2022)
558. Yan, S., Zou, X., Ilkhani, M., Jones, A.: An efficient multiscale surrogate modelling framework for composite materials considering progressive damage based on artificial neural networks. *Composites Part B: Engineering* **194**, 108014 (2020)
559. Yáñez-Márquez, C.: Toward the bleaching of the black boxes: Minimalist machine learning. *IT Professional* **22**(4), 51–56 (2020)
560. Yang, C., Kim, Y., Ryu, S., Gu, G.X.: Prediction of composite microstructure stress-strain curves using convolutional neural networks. *Materials & Design* **189**, 108509 (2020)
561. Yang, L., Meng, X., Karniadakis, G.E.: B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data. *Journal of Computational Physics* **425**, 109913 (2021)
562. Yang, L., Zhang, D., Karniadakis, G.E.: Physics-informed generative adversarial networks for stochastic differential equations. *SIAM Journal on Scientific Computing* **42**(1), A292–A317 (2020)
563. Ye, W., Hohl, J., Mushongera, L.T.: Prediction of cyclic damage in metallic alloys with crystal plasticity modeling enhanced by machine learning. *Materialia* **22**, 101388 (2022)
564. Ye, Y., Yang, Q., Yang, F., Huo, Y., Meng, S.: Digital twin for the structural health management of reusable spacecraft: A case study. *Engineering Fracture Mechanics* **234**, 107076 (2020)
565. Yoo, S., Lee, S., Kim, S., Hwang, K.H., Park, J.H., Kang, N.: Integrating deep learning into cad/cae system: generative design and evaluation of 3D conceptual wheel. *Structural and Multidisciplinary Optimization* pp. 1–23 (2021)
566. Yu, S., Tack, J., Mo, S., Kim, H., Kim, J., Ha, J.W., Shin, J.: Generating videos with dynamics-aware implicit generative adversarial networks. *arXiv preprint arXiv:2202.10571* (2022)
567. Yu, Y., Hur, T., Jung, J., Jang, I.G.: Deep learning for determining a near-optimal topological design without any iteration. *Structural and Multidisciplinary Optimization* **59**(3), 787–799 (2019)
568. Yu, Y., Rashidi, M., Samali, B., Yousefi, A.M., Wang, W.: Multi-image-feature-based hierarchical concrete crack identification framework using optimized svm multi-classifiers and d–s fusion algorithm for bridge structures. *Remote Sensing* **13**(2), 240 (2021)
569. Yuan, D., Gu, C., Wei, B., Qin, X., Xu, W.: A high-performance displacement prediction model of concrete dams integrating signal processing and multiple machine learning techniques. *Applied Mathematical Modelling* (2022)
570. Yuan, F.G., Zargar, S.A., Chen, Q., Wang, S.: Machine learning for structural health monitoring: challenges and opportunities. *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2020* **11379**, 1137903 (2020)
571. Yucel, M., Bekdaş, G., Nigdeli, S.M., Sevgen, S.: Estimation of optimum tuned mass damper parameters via machine learning. *Journal of Building Engineering* **26**, 100847 (2019)
572. Yuvaraj, P., Murthy, A.R., Iyer, N.R., Sekar, S., Samui, P.: Support vector regression based models to predict fracture characteristics of high strength and ultra high strength concrete beams. *Engineering Fracture Mechanics* **98**, 29–43 (2013)
573. Yvonnet, J., He, Q.C.: The reduced model multiscale method (R3M) for the non-linear homogenization of hyperelastic media at finite strains. *Journal of Computational Physics* **223**(1), 341–368 (2007)
574. Yvonnet, J., Monteiro, E., He, Q.C.: Computational homogenization method and reduced database model for hyperelastic heterogeneous structures. *International Journal for Multiscale Computational Engineering* **11**(3) (2013)
575. Zadpoor, A.A.: Mechanical meta-materials. *Materials Horizons* **3**(5), 371–381 (2016)
576. Zehtaban, L., Elazhary, O., Roller, D.: A framework for similarity recognition of CAD models. *Journal of Computational Design and Engineering* **3**(3), 274–285 (2016)
577. Zhan, Z., Li, H.: A novel approach based on the elastoplastic fatigue damage and machine learning models for life prediction of aerospace alloy parts fabricated by additive manufacturing. *International Journal of Fatigue* **145**, 106089 (2021)

578. Zhang, D., Guo, L., Karniadakis, G.E.: Learning in modal space: Solving time-dependent stochastic PDEs using physics-informed neural networks. *SIAM Journal on Scientific Computing* **42**(2), A639–A665 (2020)
579. Zhang, J., Sato, T., Iai, S.: Novel support vector regression for structural system identification. *Structural Control and Health Monitoring: The Official Journal of the International Association for Structural Control and Monitoring and of the European Association for the Control of Structures* **14**(4), 609–626 (2007)
580. Zhang, P., Yin, Z.Y., Jin, Y.F.: State-of-the-art review of machine learning applications in constitutive modeling of soils. *Archives of Computational Methods in Engineering* **28**(5), 3661–3686 (2021)
581. Zhang, W., Mehta, A., Desai, P.S., Higgs III, C.F.: Machine learning enabled powder spreading process map for metal additive manufacturing (am). In: 2017 International Solid Freeform Fabrication Symposium. University of Texas at Austin (2017)
582. Zhang, X.L., Michelén-Ströfer, C., Xiao, H.: Regularized ensemble Kalman methods for inverse problems. *Journal of Computational Physics* **416**, 109517 (2020)
583. Zhang, Z., Friedrich, K.: Artificial neural networks applied to polymer composites: a review. *Composites Science and Technology* **63**(14), 2029–2044 (2003)
584. Zhang, Z., Hsu, T.Y., Wei, H.H., Chen, J.H.: Development of a data-mining technique for regional-scale evaluation of building seismic vulnerability. *Applied Sciences* **9**(7), 1502 (2019)
585. Zhang, Z., Liu, Y.: Robust data-driven discovery of partial differential equations under uncertainties. arXiv preprint arXiv:2102.06504 (2021)
586. Zhao, P., Liao, W., Xue, H., Lu, X.: Intelligent design method for beam and slab of shear wall structure based on deep learning. *Journal of Building Engineering* **57**, 104838 (2022)
587. Zhao, Y., Akolekar, H.D., Weatheritt, J., Michelassi, V., Sandberg, R.D.: RANS turbulence model development using CFD-driven machine learning. *Journal of Computational Physics* **411**, 109413 (2020)
588. Zheng, B., Yang, J., Liang, B., Cheng, J.c.: Inverse design of acoustic metamaterials based on machine learning using a gauss–bayesian model. *Journal of Applied Physics* **128**(13), 134902 (2020)
589. Zheng, H., Moosavi, V., Akbarzadeh, M.: Machine learning assisted evaluations in structural design and construction. *Automation in Construction* **119**, 103346 (2020)
590. Zheng, X., Zheng, P., Zheng, L., Zhang, Y., Zhang, R.Z.: Multi-channel convolutional neural networks for materials properties prediction. *Computational Materials Science* **173**, 109436 (2020)
591. Zhu, Y., Zabaras, N., Koutsourelakis, P.S., Perdikaris, P.: Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data. *Journal of Computational Physics* **394**, 56–81 (2019)
592. Zhuang, X., Guo, H., Alajlan, N., Zhu, H., Rabczuk, T.: Deep autoencoder based energy method for the bending, vibration, and buckling analysis of Kirchhoff plates with transfer learning. *European Journal of Mechanics-A/Solids* **87**, 104225 (2021)
593. Zou, H., Hastie, T.: Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **67**(2), 301–320 (2005)

Acronyms

AI	Artificial Intelligence.
AIC	Akaike Information Criterion.
AM	Additive Manufacturing.
BNN	Bayesian Neural Network.
BRNN	Bidirectional Recurrent Neural Network.
BRT	Boosted Regression Trees.
CAE	Computer Aided Engineering.
CART	Classification And Regresssion Tree.
CNN	Convolutional Neural Network.
DBPIE	Data-Based Physics-Informed Engineering.
DDDAS	Dynamic Data-Driven Application Systems.
DED	Direct Energy Deposition.
DL	Deep Learning.
DM	Data Mining.
DMD	Dynamic Mode Decomposition.
DT	Decision Trees.
EBM	Electron Beam Melting.
FDA	Flexible Discriminant Analysis.
FDM	Fused Deposition Modeling.
FE2	Finite Element based computational homogenization (FE squared).
FEMa	Finite Element Machine.
FENA	Finite Element Network Analysis.
FFNN	Feed Forward Neural Network.
GA	Genetic Algorithm.
GAN	Generative Adversarial Network.
GPR	Gaussian Process Regression.
GPU	Graphic Processing Unit.
GRU	Gated Recurrent Unit.
ID3	Iterative Dichotomiser 3.
ISOMAP	Isometric Mapping.
JIT	Just-In-Time.
KL	Kullback-Leibler.
k-NN	<i>k</i> -Nearest Neighbours.
kPCA	Kernel (Nonlinear) Principal Component Analysis.

LASSO	Least Absolute Shrinkage Selection Operator, sparse or L1 regularization.
LATIN	LArge Time INcrement.
LDA	Linear (Fisher's) Discriminant Analysis.
LES	Large Eddy Simulations.
LLE	Local Linear Embedding.
LOOCV	Leave-one-out Cross Validation.
LS-SVM	Least Squares Support Vector Machines.
LSTM	Long Short-Term Memory.
MAC	Modal Assurance Criterion.
MARS	Multivariate Adaptive Regression Spline.
MCDM	Multicriteria Decision Making.
ML	Machine Learning.
MOR	Model Order Reduction.
NB	Naïve Bayes.
NBT	Naïve Bayes Trees.
NEXP	Numerically EXplicit Potentials.
PBDS	Physics Based Data Science.
PCA	Principal Component Analysis.
PGD	Proper Generalized Decomposition.
PGML	Physics Guided Machine Learning.
PIDS	Physics Informed Data Science.
PIML	Physics Informed Machine Learning.
PiSL	Physics Informed Spline Learning.
POD	Proper Orthogonal Decomposition.
QDA	Quadratic Discriminant Analysis.
RANS	Reynolds-Averaged Navier Stokes.
RBF	Radial Basis Function.
RC	Reinforced Concrete.
RDA	Regularized Discriminant Analysis.
RF	Random Forest.
RVE	Representative Volume Element.
SHM	Structural Health Monitoring.
SINDy	Sparse Identification of Nonlinear Dynamics.
SLM	Selective Laser Melting.
SSI	Structural System Identification.
STRidge	Sequential Threshold Ridge regression.
SVD	Singular Value Decomposition.
SVM	Support Vector Machine.
SVR	Support Vector Regression.
TDA	Topological Data Analysis.
TO	Topology Optimization.
UAV	Unmanned Aerial Vehicle.
UMAP	Uniform Manifold Approximation and Projection.
WENO-NN	Weighted Essentially Non-Oscillatory Neural Network.
WT	Wind Turbines.
XAI	eXplainable Artificial Intelligence.