Structural and Multidisciplinary Optimization (2022) 65: 266
https://doi.org/10.1007/s00158-022-03369-9

REVIEW PAPER

A survey of machine learning techniques in structural and multidisciplinary optimization

Palaniappan Ramu[1] · Pugazhenthi Thananjayan[1] · Erdem Acar[2] · Gamze Bayrak[2] · Jeong Woo Park[3] · Ikjin Lee[3]

Received: 30 December 2021 / Revised: 10 August 2022 / Accepted: 15 August 2022 / Published online: 10 September 2022

Abstract
Machine Learning (ML) techniques have been used in an extensive range of applications in the field of structural and multidisciplinary optimization over the last few years. This paper presents a survey of this wide but disjointed literature on ML techniques in the structural and multidisciplinary optimization field. First, we discuss the challenges associated with conventional optimization and how Machine learning can address them. Then, we review the literature in the context of how ML can accelerate design synthesis and optimization. Some real-life engineering applications in structural design, material design, fluid mechanics, aerodynamics, heat transfer, and multidisciplinary design are summarized, and a brief list of widely used open-source codes as well as commercial packages are provided. Finally, the survey culminates with some concluding remarks and future research suggestions. For the sake of completeness, categories of ML problems, algorithms, and paradigms are presented in the Appendix.

Keywords Classification · Clustering · Deep learning · Design diversity · Dimension reduction · Generative design · Machine learning · Neural network · Optimization · Reinforcement learning · Regression · Supervised/unsupervised learning · Uncertainty · Variational autoencoder

1 Introduction

1.1 Rise of machine learning (ML)

There has been explosive growth in the application of ML to numerous areas of engineering disciplines including structural design, material design, robotics, fluid mechanics, and heat transfer. The ASME Journal of Mechanical Design published a special issue on "Machine Learning for Engineering Design" in 2019 [Panchal et al. (2019)] consisting of 24 papers. There exists a wide but disjointed literature on the application of ML in engineering, especially structural and multidisciplinary optimization applications. In the past decades, various studies using ML have been published in the journal of Structural and Multidisciplinary Optimization (SAMO), but no articles have reviewed them. For an interested reader, it can be difficult to pinpoint useful articles in this large number of studies. Hence, we attempt to trace the growth of ML techniques in the context of optimization, while our focus is mainly on the research published in the last decade. As depicted in Fig. 1, one can a phenomenal increase in the number of ML articles published in SAMO since 2010. Publish or Perish software [Harzing (2007)] and the Google Scholar database was used to generate the data. The keywords used were 'design optimization', 'ML', 'AI', and 'surrogates'.

1.2 Challenges in structural and multidisciplinary optimization

Structural and multidisciplinary optimization is a methodology used to design structural systems which have strong interaction with other disciplines such as fluids, thermal, and

Responsible Editor: Byeng D. Youn

✉ Ikjin Lee
ikjin.lee@kaist.ac.kr

[1] Advanced Design, Optimization and Probabilistic Techniques Laboratory, Department of Engineering Design, Indian Institute of Technology Madras, Chennai 600036, India

[2] Department of Mechanical Engineering, TOBB University of Economics and Technology, 06560 Ankara, Turkey

[3] Mechanical Engineering Department, Korea Advanced Institute of Science and Technology, 291 Daehak-ro, Yuseong-gu, Daejeon 34141, Republic of Korea

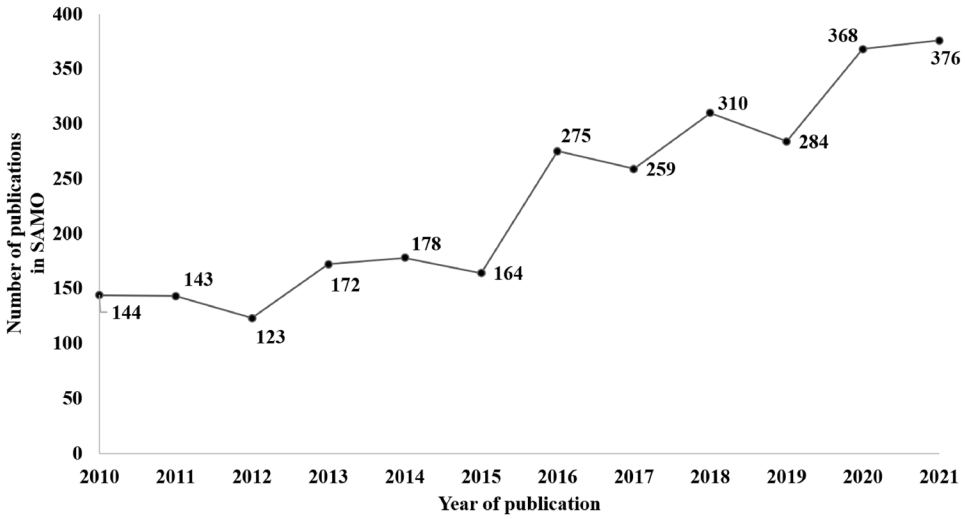


Fig. 1 Growth of ML publications in the journal of Structural and Multidisciplinary Optimization (SAMO) from 2010 until 2021

Table 1 Keyword frequency in the SAMO papers published during 2017 ~ 2021[a]

Keywords	Frequency	Covered Section
Optimization/Shape/Manufacturing	1,279/107/83	Sections 3 & 4
Topology	539	Section 3
Design/Structure/Material	439/359/145	Section 5.1
Reliability/Uncertainty	222/140	Section 2.5
Kriging/Surrogate/Model	117/116/317	Sections 2.1, 2.3 & 2.4
Finite element	57	Section 2.2

[a]A total of 1,432 papers & 6,855 keywords

manufacturing, among others. The designers need to optimize several design variables simultaneously from several disciplines. The main challenges pertaining to structural and multidisciplinary optimization can be broadly classified as computational and organizational. The computational challenges are obvious since multiple disciplines involve a lot of design variables and thus increase the dimensionality of the design and analysis. Each discipline itself might involve linear and nonlinear analysis and at the system level might require solving a more expensive multi-objective optimization formulation. The organizational challenges involve the integration and interoperability of data and models between multiple solvers such as structural, fluid flow, reliability, and cost analysis among others. Table 1 shows the keyword frequency of SAMO articles published during the 2017–2021 time period. In Table 1, generic keywords such as 'based', 'method', and 'analysis' are removed. Following are the detailed challenges related to the keywords:

- Conventional surrogate modeling such as Kriging, and neural network (NN), model reduction techniques such as active subspace, and Grassmann manifold demand large data obtained from expensive computer simulations for training to predict accurately because of the nonlinear characteristic of the model is not known.
- Although a large number of data is available, conventional surrogate modeling including multi-fidelity surrogate (MFS) modeling may cause an overfitting problem unless appropriate basis functions and parameters are selected.
- Due to the curse of dimensionality, conventional surrogate modeling and uncertainty analysis may have difficulty maintaining high accuracy for high dimensional problems.
- Conventional design optimization typically does not reuse existing prior information obtained from previous iterations or optimization results to solve similar design configurations.
- Conventional design optimization requires a significant amount of iterations to obtain the final design. Thus, its convergence – especially, in topology optimization (TO), is very slow.
- The conventional finite element method (FEM) strives to better the generalization capabilities and choice of parameters to obtain accurate solutions when the boundary conditions are discontinuous and computational cost involved to solve large complex problems even with minor modifications and estimation of sensitivities.

ML-based approaches can help address the existing challenges mentioned above by their inherent capability to learn from existing prior information and to classify patterns. Hence, this article aims to review the structural and multidisciplinary optimization applications of ML studies under this general approach. Furthermore, practical engineering applications and software tools are listed in this paper so that

an interested reader can easily understand where to apply an ML algorithm or what software to use.

1.3 Organization of the paper

The remainder of the paper is organized as follows. As described in Fig. 2; Sections 2 and 3 present a review of studies in which ML has been used to facilitate design optimization in terms of efficiency and accuracy. For this purpose, ML is mainly utilized to generate a surrogate model including MFS or to accelerate uncertainty analysis and TO. Section 4 provides a review of ML for design synthesis where design diversity, improvement of design quality and performance, and human preference studies, which are categorized according to the purpose of design synthesis, are reviewed. In this section, ML is utilized not to facilitate design optimization but to directly generate final designs. Section 5 reviews engineering applications in various engineering fields, including structures and materials, and introduces software tools for ML. Section 6 provides a summary with concluding remarks on ML in structural and multidisciplinary design optimization. Finally, categories of ML problems and covers the basic concept of various ML methods widely used in design optimization can be found in the Appendix.

2 Machine learning to facilitate design and optimization

Design optimization has been an active field of research for the last four decades. The top-level goals addressed by current research include: (i) reducing computational burden or speeding the optimization process, (ii) making different modules (such as the design of experiments (DoE)) of optimization more efficient and developing approaches to solve nonlinear, stochastic or ill-posed problems faster and (iii) generalization. In the following, we discuss the instances where ML techniques address the above goals and later discuss related developments in detail. In terms of (i), as early as 1992, [Cerbone (1992)] introduced a novel three-step ML framework to speed up numerical optimization using symbolic learning, inductive learning algorithms, and inductive discovery techniques to reduce the number of independent variables. This methodology was implemented in 2D structural design yielding 95% speedup over the classical optimization methods. [Freiesleben et al. (2020)], in the context of quality management methodologies in production, investigated the role of DoE amidst the surge in ML approaches, addressing (ii). The authors accessed five core parts of DoE for the applicability of ML. They observed that DoE can be made more effective and efficient by the application of ML. Data availability, data quality, insufficient data,

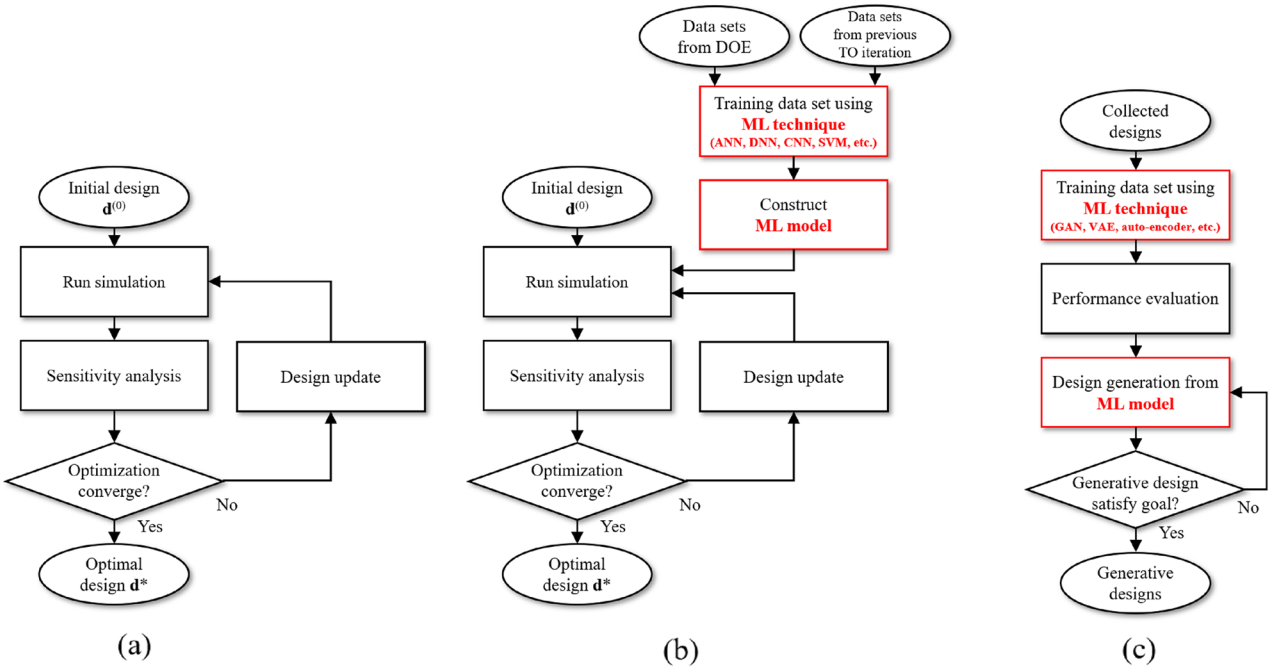


Fig. 2 Comparison between general optimization and optimization assisted by ML: **a** general optimization, **b** ML to facilitate structural & topology optimization (Sects. 2 and 3), and **c** ML for generative design (Sect. 4)

computational resources, and overfitting and biasing in ML are some of the aspects that need attention.

Surrogates are widely used to reduce computational burden addressing (i). [Adeli and Park (1995)] introduced the first known work that uses NN and the concept of learning in structural optimization. They used the Karush–Kuhn–Tucker (KKT) necessary condition as the learning rule while formulating a neural dynamics model for structural optimization problems. They demonstrated global convergence and the ability to parallel processes. While conventional surrogate approaches are usually treated as a data-driven approach, recently, in addition to the available data, physics-embedded ML [Karniadakis et al. (2021)] techniques have been used to solve the partial differential equation (PDE) and are suitable for ill-posed and inverse problems, addressing (ii). For similar optimization problems, techniques such as reinforcement learning (RL) are highly generalizable to unseen system configurations, addressing (iii). That is, in the context of design, sometimes the requirements are often similar but slightly different requiring one to solve an entirely new optimization problem owing to unseen configurations. [Yonekura and Hattori (2019)] developed a framework for design optimization using deep RL where they train an agent in advance to find the optimal solution for slightly different requirements. Similar work includes [Odonkor and Lewis (2019)]'s RL-based data-driven model to design temporal arbitrage policies for the operation of shared energy storage and photovoltaic systems.

Figure 3 shows the tree structure of ML to facilitate design optimization aimed at reducing computational cost can be classified as in Sects. 2.1–2.4 according to the problem to be solved, and studies dealing with uncertainty are in Sect. 2.5.

2.1 Surrogate modeling

Surrogates or metamodels [Forrester et al. (2008)] are function approximation techniques that fit functions to data. The fitted functions replace expensive computer simulations or physical experiments, to obtain responses at new data points. Traditionally, researchers have used techniques such as the polynomial response surface approach, Kriging, and radial basis function and their ensemble as surrogates [Goel et al. (2007); Acar and Rais-Rohani (2009)]. In the recent past, ML techniques have also been used as metamodels. The precursor to ML in surrogates or approximations

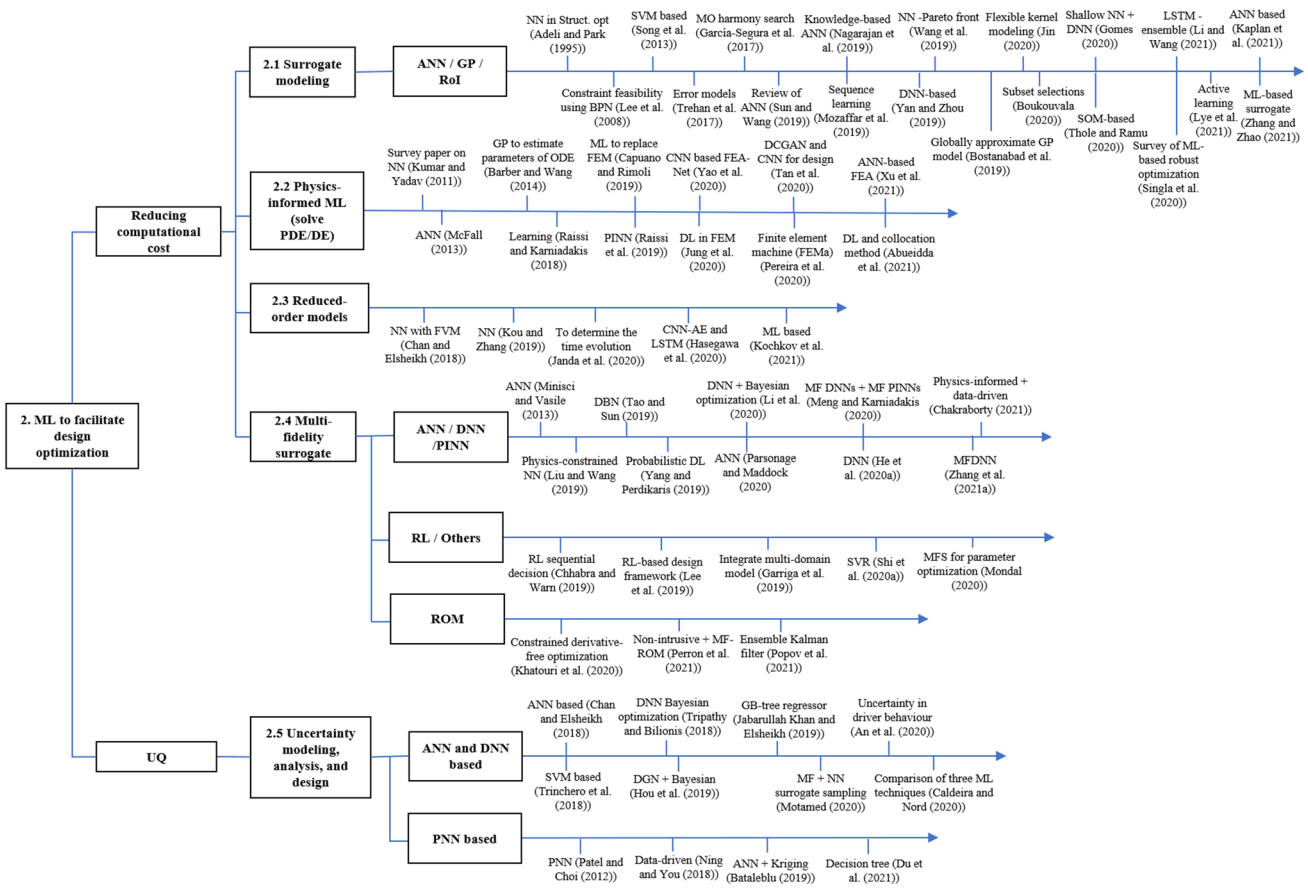


Fig. 3 Tree structure of ML to facilitate design optimization

is predominantly a NN model, which was attractive as an unsupervised learning technique for point estimates. Over the years, the learning aspect of NNs received attention, and NNs were combined with both gradient- and heuristic-based approaches to perform optimization. ML approaches, of which NN is a subset, became a logical extension because of the variety of problems they could address. ML approaches, for instance, have taken advantage of the learning aspect and permitted bypassing the expensive simulations, in the context of design and analysis. A surge in the computational ecosystem allowed the development of deep neural network (DNN) approaches as surrogates. Recent developments involve including the physics of the problem as prior information into a NN model to solve more complex problems in less time and with limited data.

[Lee et al. (2008)] discussed the implementation of back propagation neural networks (BPN)-based metamodels that ensure the constraint feasibility. It was shown that genetic algorithm-based learning approaches are more efficient than derivative-based approaches. It was also noted that the proposed approach can be extended for other metamodel techniques such as Kriging and support vector machine (SVM) as well. [García-Segura et al. (2017)]integrated multi-objective harmony search with artificial neural networks (ANNs) to reduce the computational time required for finite element analysis (FEA). The proposed methodology was verified using the post-tensioned concrete box-girder road bridge multi-objective problem. [Nagarajan et al. (2019)] developed a knowledge-based ANN that combines the dimensional analysis conceptual modeling and typical ANN. The hybrid approach had topological zones derived from the process knowledge augmented by ANNs where knowledge was missing. The proposed approach had better generalization capabilities compared to classical ANN. [Sun and Wang (2019)] reviewed ANN-based surrogate modeling for different objectives of aerodynamic design applications. This work discussed the selection of the type of NN that may have an influence on the optimization effect. [Wang et al. (2019)] introduced a new approach where, at each iteration, the NN surrogate was used to propose new trial solutions in multi-objective optimization. It was observed that the Pareto front converges more quickly than the genetic and particle swarm algorithms. [Mozaffar et al. (2019)] developed a sequence learning scheme to find the constitutive law that links inputs and outputs. In the process, they bypass finding equations for materials with complex microstructures and trace the yield surface evolution iteratively. [Gomes (2020)] applied shallow NNs and DNNs as the metamodels in structural reliability analysis, using adaptive experimental designs. Two linear activation functions were used for both types of ANNs: tansig for shallow NNs, and rectifier linear unit (ReLU) for DNNs. Though the DNN was outperformed, nevertheless the performances of shallow ANNs were comparably acceptable with those of other metamodels. [Zhang and Zhao (2021)] introduced an ML-based surrogate method for the distributed fluid system. Dimensionality reduction was used to reduce the flow field dimension, and NN was used as a regression model. In the work, the whole fluid domain is subdivided into subdomains and then a surrogate model for each subdomain is generated by treating both boundary information and distributed flow parameters as the input parameters. [Kaplan et al. (2021)] used ANNs to predict the structural response of a store externally carried by a jet fighter. [Trehan et al. (2017)] introduced a general method for error modeling of surrogate models of dynamical systems via ML. The authors proposed four error modeling techniques, namely quantity of interest error, the relative quantity of interest error, state error, and relative state error. The proposed method was applied to the subsurface flow problem and performed the best using the proposed error-modeling approach.

Gaussian process (GP) [Williams and Rasmussen (2006)] models have been used in interpolation mode or regression mode as surrogate models as precursors to generative models. Researchers have used GP-based surrogate models in various studies including damage prediction of aluminum parts [Solanki et al. (2009)], vehicle crashworthiness [Acar and Solanki (2009)], nuclear engineering applications [Baraldi et al. (2015)], structural reliability analysis [Su et al. (2017)] as well as fuel performance and thermo-hydraulics [Wu et al. (2018)]. [Bostanabad et al. (2019)] proposed a novel globally approximate GP to enable the GP modeling of big datasets. They introduced an intuitive method of building a collection of independent GPs that use the converged hyperparameters as their hyperparameters. The globally approximate GP method was implemented in a data-driven metamaterials unit cell design process to estimate the desired elastic property. The complex material design problem was transferred into a parametric one using spectral descriptors. [Jin (2020)] proposed a methodology to choose a proper kernel covariance function of GPR. The proposed method worked better than the methods in both interpolation and regression problems. Flexible kernel modeling of the proposed method was able to capture the functional relationships by adaptively constructing the suitable kernels, and thus it can be implemented in surrogate modeling and predictive modeling using GPR. [Kim and Boukouvala (2020)] compared five different subset selections for regression techniques for surrogate modeling. They investigated the subset selection for regression compared to the complex kernel-based interpolating metamodels for better data-driven optimization. Standard benchmark problems up to 30D were used to compare the performance of the proposed method with the Kriging surrogate model. The regression surrogate function performed well in low-dimension problems while Kriging was outperformed in high-dimension cases. [Singla

et al. (2020)] presented a survey of robust optimization in the context of [ML specific to SVM or support vector regression (SVR). They discussed the robustness of algorithms when the data has uncertainties. Li and Wang (2021)] developed a long short term memory (LSTM) -based ensemble learning framework for time-dependent reliability analysis involving stochastic processes. Monte Carlo simulation (MCS) is adopted for estimating the time-dependent reliability based on the coupling of the LSTM network with the GP modeling technique. The LSTM network was employed to learn the relationship between the stochastic processes and time-dependent system responses, and conditional limit-state functions were introduced by fixing the time-independent random variables.

Often surrogate models are used to understand regions of interest (RoI) and permit adaptive sampling in the RoI. [Song et al. (2013)] introduced an efficient classification methodology using a virtual SVM which was used to generate adaptive samples to improve the accuracy of the decision function for highly non-linear problems. The samples near the limit-state function that are chosen by the sequential sampling strategy were integrated with virtual SVM. [Yan and Zhou (2019)] presented a DNN-based adaptive MFS modeling framework for large-scale Bayesian inverse problems. The key idea is to construct an offline DNN-based surrogate model according to the prior distribution, and refine the DNN-surrogate model adaptively and locally using fewer high-fidelity simulations. An adaptive procedure was framed to correct the surrogate within the Markov chain Monte Carlo framework. [Thole and Ramu (2020)] proposed modified self-organizing maps whose maps are inherently interpretable permitting visualization of design space. The self-organizing map is a type of NN and allows the identifying region of interest for targeted sampling leading to accurate surrogates. [Lye et al. (2021)] presented a novel active learning procedure where DNNs are used as efficient surrogates for PDEs, particularly for approximating observables of PDEs. In this regard, they proposed a DNNopt algorithm that combines quasi-Newton with a DNN surrogate but converged slowly for high-dimensional problems. Hence, an iterative surrogate model optimization (ISMO) algorithm was proposed where the key idea is to iteratively augment the training set for a sequence of DNNs. In terms of computing power and robustness, the ISMO algorithm performed better than the DNNopt. Though surrogates are cheaper to evaluate compared to the original model evaluation, building surrogates which is called 'training', requires a large sample, especially for non-linear problems. Hence, researchers started focusing on using additional available information to reduce computational expense.

2.2 Physics-informed ML

[Raissi and Karniadakis (2018)] proposed a new framework for learning PDEs from small data. They introduced a hidden physics model that takes advantage of the underlying physical laws and/or governing equations to extract the information from small data generated from highly complex systems. The proposed framework can be useful in cases where the learning of experimental data is noisy but the governing equation is known. The performance of the proposed approach was showcased through a variety of problems such as Navier–Stokes, Schrödinger, Kuramoto–Sivashinsky, and time-dependent linear fractional equations without requiring large experimental data. [Raissi et al. (2019)] introduced physics-informed NN (PINN) that is trained to solve supervised learning tasks with underlying laws of physics. They presented two main classes of subproblems namely data-driven solutions and data-driven discovery of PDEs. The proposed method was demonstrated with several problems in fluids, quantum mechanics, and reaction–diffusion systems. However, the authors noted that the proposed method should not be viewed as a replacement for classical numerical methods (finite elements, spectra methods, etc.) for solving PDEs. [Li and Mei (2021)] introduced an improved method for PDE by NNs. The physics of the PDE are introduced as regularization into the NNs. This idea enables the use of limited data to learn solutions of PDE by the NN. The discretization of PDE is avoided by randomly selected samples in the space–time domain to train the PINN. The performance of the proposed methodology is demonstrated by the wave equations, the KdV-Burgers equation, and the KdV equation showing that the PINN is more effective in solving PDEs. [Tan et al. (2020)] proposed a deep learning (DL) model that combines both deep convolutional generative adversarial network (DCGAN) and convolutional neural network (CNN) for microstructural materials design. The geometrical constraints were satisfied by the DCGAN and were used to generate the design, while the CNN was used as the surrogate model that connects the microstructure to its properties. The performance of the developed framework was exhibited on microstructural materials design with intended compliance tensor, subject to geometrical constraints. The error in the predicted compliance tensor was less than 5% and thus results were more accurate than the traditional FEA. [Abueidda et al. (2021)] developed a deep collocation approach which is a combination of DL and collocation method to solve PDE by describing structures' deformation.

[Kumar and Yadav (2011)] surveyed and classified different multi-layer-perceptron and radial basis function NNs to solve differential equations. The radial basis function NN technique provided a more accurate solution than the multi-layer-perceptron. Compared to FEM, the NN provided very good generalization properties. [McFall (2013)] introduced

an automated design parameter selection process to choose a single ANN from an ensemble comprising numerous combinations of design parameters, random starting weights, and biases to solve differential equations with discontinuous boundary conditions. Three different thermal-fluid science examples along with Navier–Stokes equations were used to verify the efficacy of the proposed methodology, and the resulting approximations provided low solution error. [Barber and Wang (2014)] propose a generative model that links state derivative information to system observations using a GP for estimating parameters in coupled ordinary differential equation (ODE). They used a simple belief network that consists of the states, their derivatives, observations, and related parameters. The proposed GP-ODE was amenable to alternative approximation techniques such as variational approximations. Benchmark problems were solved using the proposed method and performed significantly better than other GPs. [Capuano and Rimoli (2019)] formulated a methodology that uses the capability of ML to avoid the need of finding internal displacement fields thus eradicating the need for numerical iterations. The data for training the machine model was extracted from the existing FEA thus producing the approximate model called a smart element. The proposed method was demonstrated with 3D truss and 2D multiscale structure problems, and the computational cost was reduced without compromising the accuracy of a variety of complex finite element formulations. [Jung et al. (2020)] introduced a methodology to generate the stiffness matrix of finite elements using DL and developed 8- and 4-noded quadrilateral elements. This study involved and introduced new concepts such as normalized element geometry, the reference data model for the training data, pre-processing for the input, and post-processing for the output. The performance of DL4 elements and DL8 elements was evaluated through various numerical examples and DL8 elements performed well in terms of accuracy and computational cost. By reducing the number of weights and optimizing the structure of the network, the computational efficiency can be easily improved. [Pereira et al. (2020)] introduced a new framework called finite element machine (FEMa) for the design of pattern classifiers based on FEA. FEMa can learn a probabilistic manifold constructed over the training data, and FEMa can be parameterless, which is the heart of a finite element basis. The proposed methodology can obtain very competitive results when compared to some other supervised pattern recognition techniques such as k-nearest neighbors, SVM, and ANN. [Yao et al. (2020)] proposed a hybrid deep CNN called FEA-Net that takes advantage of different prior knowledge about the physics of the problem to build data-driven models. The proposed network was used to predict the mechanical response of the system under external loading with limited training data samples. The proposed method is demonstrated in multi-physics and multiphase problems. [Xu et al. (2021)] proposed an ANN-based optimization process to estimate the objective function and analytical sensitivities of FEA. This process includes a sampling strategy to improve efficiency and regression accuracy and is compared with a conventional FEA-based discrete material optimization method for validation. Specifically, an 84% computational time savings in the small-scale optimization of the carbon fiber reinforced plastic engine hood problem was achieved.

2.3 Reduced-order models (ROM)

Though ROMs are computationally inexpensive representations of real-time simulation models, constructing ROMs is not cheap. In addition, ROMs are not robust to parameter changes. [Amsallem and Farhat (2008)] address both these issues by using an interpolation method in tangent space to a Grassmann manifold. The proposed method is illustrated with computational-fluid-dynamics-based aeroelastic reduced-order models of two realistic aircraft configurations. [Kou and Zhang (2019)] developed a hybrid reduced-order model to simulate linear and nonlinear aerodynamics. The physics of the unsteady flow is introduced by constructing a hybrid parallel system consisting of a linear autoregressive with an exogenous input model and a nonlinear NN model. The framework can incorporate both linear and nonlinear system identification. [Chan and Elsheikh (2018)] combined the feed-forward NN with the multiscale finite volume method (MsFV) to form a data-driven approach to solving problems using coarse-scale basis functions. NN is used as a surrogate and thus learned to generate subsequent basis functions at a cheaper computational cost. The proposed method showed promising results compared to traditional finite volume methods (FVM) and MsFV. [Pawar et al. (2019)] introduced a modular DNN framework for data-driven reduced-order modeling of dynamical systems relevant to fluid flows. They proposed various DNN architectures which numerically predict the evolution of dynamical systems by learning from either using discrete state or slope information of the system. [Janda et al. (2020)] generated a ROM for an unconstrained glass plate exposed to low-velocity impact. Combined with the Hertzian theory of non-adhesive contact, the ROM is used to determine the time evolution of contact forces for arbitrary mass, stiffness, and initial velocity of the impactor. [Hasegawa et al. (2020)] investigated the capability of an ML-based ROM for two-dimensional unsteady flows around a circular cylinder at different Reynolds numbers. First, CNN-AE was utilized to map high-dimensional flow fields obtained by direct numerical simulation into a low-dimensional latent space while keeping their spatially coherent information. Then, an LSTM was trained to learn the temporal evolution of the mapped latent vectors together with the information

on the Reynolds number. Using the trained LSTM model, the high-dimensional dynamics of flow fields can be reproduced with the aid of the decoder part of CNN-AE, which can map the predicted low-dimensional latent vector to the high-dimensional space. [Kochkov et al. (2021)] presented a data-driven numerical approach to accelerate the direct numerical simulation and large-eddy simulation using ML methods without compromising the accuracy. Within the traditional FDM/FVM discretization framework, the proposed method uses ML to interpolate at a coarse scale. The proposed approach results in 40- to 80-fold computational speedup for coarse grid simulation.

2.4 Multi-fidelity surrogate (MFS)

High-fidelity (HF) simulations of structural and multidisciplinary systems are often computationally expensive, and practices such as sensitivity analysis, optimization, and uncertainty modeling require the generation of multiple realizations. Even though the use of surrogate models provides an opportunity to reduce the computational burden, the number of simulations required to construct surrogate models may be intractable when the HF models are computationally expensive. Usually, less accurate but computationally inexpensive low-fidelity (LF) models are also available, and there is growing interest in using MF models that combine HF and LF models in order to achieve an acceptable level of accuracy at a reasonable cost. A comprehensive survey of MF modeling approaches can be found in the review papers of [Fernández-Godino et al. (2016)] and [Peherstorfer et al. (2018)]. The popularity of and advancements in the field of ML have resulted in growing interest in applying ML techniques to MF modeling approaches. In this paper, we focus on the recent applications of ML techniques in the MF modeling approaches.

ANNs have been widely utilized in MF modeling approaches. [Minisci and Vasile (2013)] conducted a preliminary robust design of a small/medium–scale reentry unmanned space vehicle, where ANNs were used to approximate the aerodynamic forces required. In that study, ANNs were trained and updated by means of an MF, evolution-control approach. [Parsonage and Maddock (2020)] proposed a hybrid parametric/non-parametric information correction method incorporating the sequential application of several distinct stages within an ANN-based surrogate framework, where a global MFS model was generated via a double loop ANN hyperparameter selection and training procedure.

DNNs have also been widely used in MF modeling approaches. [Tao and Sun (2019)] presented an MFS-based optimization framework where a deep belief network (DBN) was employed as the LF model, and a linear regression MFS model was established by using the DBN model and HF data. The constructed MFS model was applied to the robust optimizations for airfoil and wing under uncertainty of Mach number. [Yang and Perdikaris (2019)] presented a probabilistic DL methodology to generate MFSs for stochastic systems. The constructed MFS model was found to be successful in uncertainty propagation in high-dimensional dynamical systems. [Li et al. (2020)] proposed a DNN MF Bayesian optimization model. They stacked a set of NNs where each NN models one fidelity. In each fidelity, they fed both the original input (to the objective) and output from the previous fidelity into the NN to propagate information throughout the model. For efficient inference and tractable computation of the acquisition function, they considered the NN weights in the output layer as random variables and all the other weights as hyperparameters. He et al. (2020a) applied the DNN algorithm in fusing aerodynamic data with different fidelity levels. The MF architecture involved three fully-connected NNs that were employed to approximate LF data, and the linear part and nonlinear part of correlation for the LF and HF data, respectively. [Zhang et al. (2021a)] used MFDNN to construct a high-accuracy MFS model correlating the configuration parameters of an aircraft and its aerodynamic performance by blending different fidelity information and adaptively learning their linear or nonlinear correlation without any prior assumption. In the optimization framework, the HF model using a computational fluid dynamics (CFD) evaluation with a fine grid, and the LF model using the same CFD model with a coarse grid were applied.

Another approach in MF modeling was to utilize PINNs. [Liu and Wang (2019)] proposed an MF physics-constrained NN, where a low-cost LF physics-constrained NN was used as the baseline model, whereas a limited amount of data from an HF physics-constrained NN was used to train a second NN to predict the difference between the two models. The proposed framework was demonstrated with 2D heat transfer, phase transition, and dendritic growth problems. [Meng and Karniadakis (2020)] developed MF DNNs and MF PINNs. These models had the following properties: they could learn both the linear and nonlinear correlations adaptively, they were suitable for high-dimensional problems, they could handle inverse problems with strong nonlinearities, and they were easy to implement. [Chakraborty (2021)] presented an MF physics-informed DNN that achieved the most desirable outcome when the physics of the problem was known in an approximate sense (LF physics), and only a few HF data were available. The method blends physics-informed and data-driven DL techniques by using the concept of transfer learning. The approximate governing equation was first used to train an LF physics-informed DNN. This was followed by transfer learning where the LF model was updated by using the available HF data.

RL has also been used in MF modeling. [Chhabra and Warn (2019)] presented an RL to aid the selection of efficient

MF models when the design was viewed as a sequential decision process, and the computational cost and discriminatory power of models were unknown at the onset of the design process. [Lee et al. (2019)] formulated an RL-based design framework, which simultaneously found solutions that were more efficient compared with supervised learning approaches while using data more efficiently compared with genetic algorithm-based optimization approaches.

Even though NN has been the most commonly used ML technique used in MF modeling, other ML methods have also been used. [Shi et al. (2020a)] developed an MFS model based on SVR, where the HF and LF samples were mapped into a high-dimensional feature space through a kernel function, and then a linear model was utilized to evaluate the relationship between inputs and outputs. [Garriga et al. (2019)] integrated multi-domain models and handled libraries of systems representations at different levels of fidelities for the efficient exploration, analysis, and optimization of novel aircraft system architectures. They applied ML techniques to reduce the models to a manageable level of complexity and also to reduce the number of architectural options after the first evaluation was performed by implementing a classification algorithm.

[Mondal (2020)] discussed the MFS modeling and optimization strategies with respect to parameter optimization in additive manufacturing, model calibration, and compressor rotor design. The work also discusses hidden Markov models in the detection of thermoacoustic instabilities and lean blow-out in combustion systems using acoustic and chemiluminescence sensor data. The proposed framework achieves computational efficiency and robust predictions of regime changes with parsimonious data requirements, which deems it suitable for online applications.

Reduced-order models have also been used in MFS modeling. [Khatouri et al. (2020)] address the problem of constrained derivative-free optimization in an MF framework using Bayesian optimization techniques. They proposed to use GP models with trend functions built from the projection of LF solutions on a reduced-order basis synthesized from scarce HF snapshots. [Perron et al. (2021)] introduced a novel non-intrusive and MF ROM method based on manifold alignment. Unlike previous MFS approaches, the developed method was capable of combining fidelity levels with disparate field representations. [Popov et al. (2021)] developed a new MF ensemble Kalman filter algorithm based on a linear control variate framework. They investigated the use of reduced-order models as coarse fidelity control variates in the MF ensemble Kalman filter algorithm and provided analyses to quantify the improvements over the traditional ensemble Kalman filters.

2.5 Uncertainty modeling, analysis, and design

Due to their efficiency and accuracy in handling high-dimensional data, ML techniques have been used in uncertainty modeling, analysis, and design optimization under uncertainty. We observe that NNs, in particular probabilistic NNs (PNNs) and DNNs, have been widely used for the aforementioned purposes.

In uncertainty modeling, [Hou et al. (2019)] used deep generative networks to capture the posterior distribution in Bayesian inverse problems by learning a transport map. [Motamed (2020)] proposed an MF NN surrogate sampling method for uncertainty quantification. A two-level NN was constructed by utilizing a large set of LF data in order to accelerate the construction of an HF surrogate model with a small set of HF data. The constructed MF model was embedded in the framework of an MCS. [An et al. (2020)] focused on autonomous vehicle driving control and utilized ML classification algorithms to model and recognize the uncertainty in driver behavior. They evaluated the probability of the specific driving style based on a trained classification model. An uncertainty model based on a parameterized stochastic hybrid state chart was constructed and mapped to networks of probabilistic timed automata.

In uncertainty analysis, [Chan and Elsheikh (2018)] utilized ANNs to perform uncertainty quantification where a multiscale finite volume method was employed in the uncertainty propagation task. [Trinchero et al. (2018)] applied the SVM and the least-squares SVM regressions to the uncertainty quantification of complex systems with a high-dimensional parameter space. They found that SVM and the least-squares SVM regressions could be considered an effective solution for uncertainty quantification in high-dimensional nonlinear problems with an accuracy comparable to or even better than that of sparse polynomial chaos expansion. [Tripathy and Bilionis (2018)] parameterized the structure of a DNN such that the DNN was a composition of an encoder and one-layer perceptron. That parameterization lends the DNN surrogate the interpretation of recovering a nonlinear active subspace. They used a combination of grid searches and Bayesian global optimization to select the best set of network hyperparameters and determine the appropriate structure. [Jabarullah Khan and Elsheikh (2019)] used ML techniques (gradient boosted tree regressor) to combine the features of the MF MCS and the multi-level MCS into a single framework to reduce the computational cost of MCS for uncertainty propagation. [Caldeira and Nord (2020)] compared three different ML methods, namely Bayesian NNs, concrete dropout, and deep ensembles, in the quantification of epistemic and aleatoric uncertainties. Simulated experimental measurements of a single pendulum were utilized, and it was found that aleatoric uncertainties are modeled well by using all three methods, the deep ensemble being

the best method. However, it was also found that epistemic uncertainties were not quantified well, and in particular concrete, dropout predicted very low epistemic uncertainties.

In design optimization under uncertainty, [Patel and Choi (2012)] explored the use of PNN to facilitate the accurate estimation of probabilistic constraints in optimization problems under uncertainty. The efficiency of the proposed framework was achieved with the combination of a conventional TO method and PNN. [Ning and You (2018)] proposed a novel data-driven stochastic robust optimization framework for optimization under uncertainty leveraging labeled multi-class uncertainty data. ML methods including the Dirichlet process mixture model and maximum likelihood estimation were employed for uncertainty modeling. The robust optimization framework was further proposed based on the data-driven uncertainty model through a bi-level optimization structure. [Bataleblu (2019)] used ANNs and Kriging surrogate models in solving uncertainty-based design optimization of truss and frame structures. The author employed a novel evolution control strategy and efficient global optimization method in the framework of uncertainty-based design optimization. [Du et al. (2021)] developed a decision tree algorithm for uncertain data that can learn from uncertain data, which are characterized by interval attribute values rather than exact values. Based on the joint probability distribution of design variables, the algorithm can be constructed by evaluating each artificial splitting point. The developed method was implemented in the design of a thin-walled energy-absorbing structure subjected to crash loading to demonstrate its performance.

3 Improvement of efficiency and accuracy in topology optimization (TO)

ML has been used as a tool to improve the efficiency and accuracy of engineering design including structural optimization. In general, efficiency is compared to conventional methods in terms of computational cost, and accuracy is compared to improved design resolution, compliance, and predictive performance. In particular, various studies have been developed to improve the performance of TO, which is a popular and powerful computational approach for designing new structures, materials, and devices. However, since TO involves many design variables, expensive simulation models cannot be utilized for optimization. The tree structure in which ML is used as a tool for TO is shown in Fig. 4. ML techniques such as CNN and GAN have been applied to improve its efficiency and accuracy compared to existing methods.

3.1 Efficiency improvement using DNN

To increase the efficiency of TO, various studies using DNN as a surrogate model have been proposed. [Bi et al. (2020)] suggested a general scalable DL-based TO framework that integrates DL and parallel computing to accelerate the TO process for additive manufacturing materials. The proposed method deeply understands the iterative history data and utilizes DNN to learn an accurate and fast surrogate gradient instead of the true gradient. Since the method utilizes parallel schemes in high-performance computing, the efficiency of TO can be further improved. In additively manufactured material design for a heat conduction example, the proposed framework showed similar performance to the baseline method while reducing the computational cost by 8.6 times. [Deng et al. (2020)]introduced a method to integrate DL into stochastic optimization algorithms. In each iteration, generative simulated annealing uses DNN to evaluate the objective function, based on which new training data are generated, resulting in better accuracy than the original generative simulated annealing. The proposed algorithm was implemented in a compliance minimization problem where DNN was used to replace FEM. This algorithm reduced the computational cost to 1/300 times of the original generative simulated annealing. An interesting property that the number of function evaluations does not increase exponentially has been observed and thus the developed algorithm brought a new perspective to high-dimensional optimization. [Deng and To (2020)] proposed a new TO method where the density field of the design domain is represented by DNN. The design variables can be reduced phenomenally compared to traditional density-based optimization methods. Several 2D and 3D numerical examples with three different kernels such as Gaussian, tansig, and tribas were used to demonstrate the proposed method. In the proposed method, no filtering technique was needed and the optimal designs were free from checkerboard patterns. [Chandrasekhar and Suresh (2021)] formulated a framework for TO using NNs that directly use the NN's activation function to represent the solid isotropic material with penalization (SIMP) [Bendsøe (1989)] density field. The density field was optimized by NN's back propagation and the finite element mesh. The proposed framework was validated using several 2D and 3D benchmark problems. However, the lack of detailed features in the computed topologies and handling of distributed loads are some of the shortcomings of the method. [Chi et al. (2021)] introduced a general ML-based TO to accelerate the process for large-scale problems which allows the ML model to be trained during the TO process. The methodology significantly accelerated the TO problems with various load and boundary conditions and design requirements and can be implemented for robust TO under uncertainty.

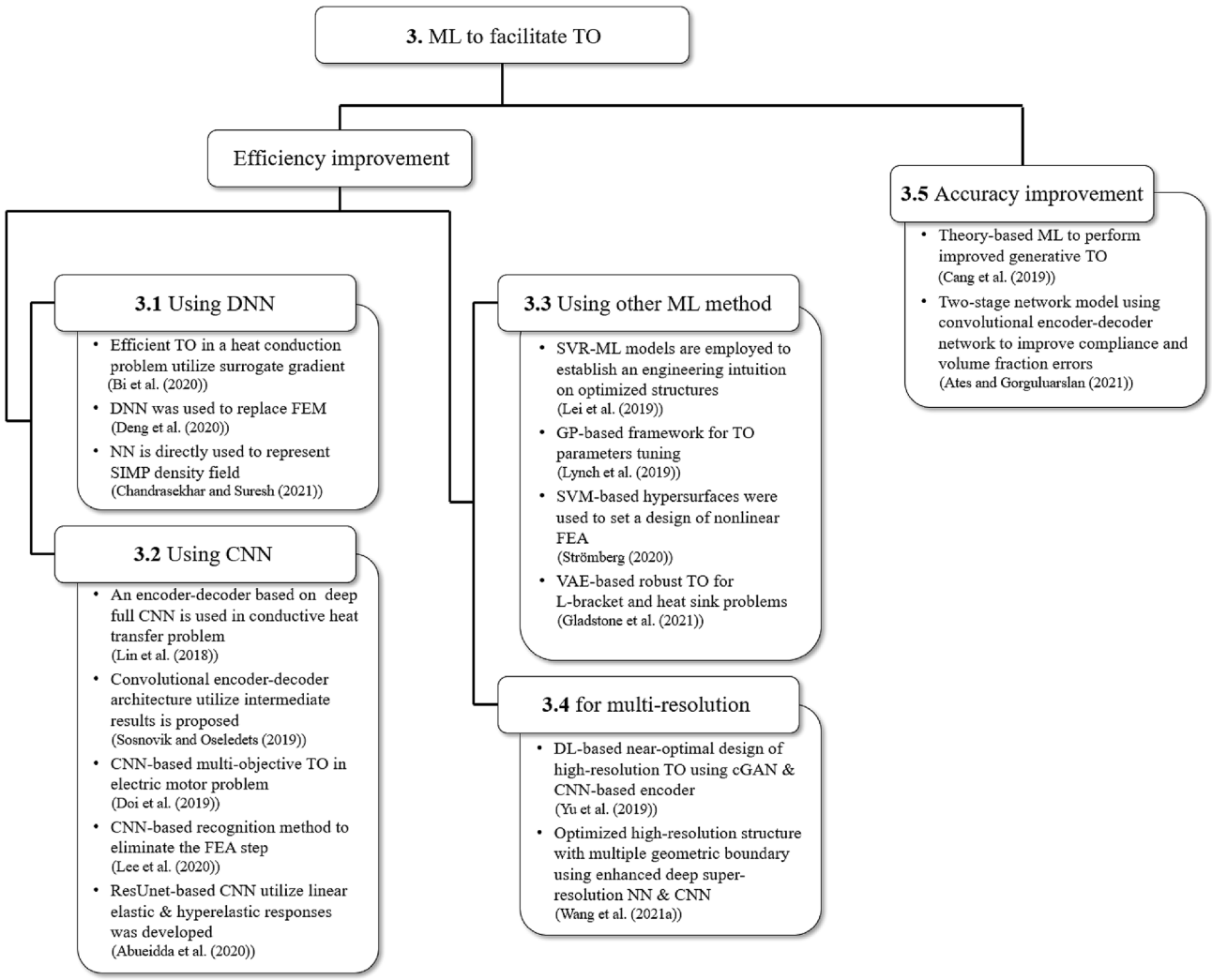


Fig. 4 Tree structure of ML to facilitate TO

3.2 Efficiency improvement using CNN

In addition, several studies related to efficiency improvement by applying CNN have also been performed. [Lin et al. (2018)] combined the DL approach with the traditional SIMP to accelerate the TO of the conductive heat transfer problem. The architecture of the proposed method is similar to the U-Net system and consists of an encoder and a decoder based on a deep full CNN. Comparing the volume-point heat conduction problem with the pure SIMP method, up to 80% reduction in computation time was achieved. [Banga et al. (2018)] proposed a data-driven DL model based on 3D encoder-decoder CNN architecture for accelerating 3D TO. The main concept of this architecture is that the final outputs of TO can be predicted from intermediate structural inputs. The models were trained from the results obtained from an FEA-based software TopOpt. The best network utilizes the density and gradient data as input to CNN. It was verified that the network with the best performance in the test set achieved 40% computational time reduction while maintaining 96% structural accuracy. [Sosnovik and Oseledets (2019)] suggested a DL-based approach to accelerate the TO. The convolutional encoder-decoder architecture and the efficient pixel-wise image labeling technique are introduced to solve the image segmentation task with high performance. The out model learned mapping from the intermediate results of the iterative method to the final structure of the design domain. Due to the early termination of SIMP, it was possible to significantly reduce the total time consumption through the experiments performed. [Lin et al. (2019)] developed a supervised DL predictor to directly infer the conductive heat transfer topologies. The physical parameters describing the cooling problem were input of the predictor. And, the architecture of the proposed method also consisted of an encoder and decoder. The accuracy of the three parallel predictors that can directly infer

the optimized topology is 96%**.** [Doi et al. (2019)] proposed a CNN-based multi-objective TO. Cross-sectional images and performance of electric motors obtained during multi-objective TO based on FEM and genetic algorithms were used for training the CNN. In the two procedures according to the training method of the proposed approach, the amount of FE computations was reduced to 30% and 50%, respectively. [Sasaki and Igarashi (2019a, b)] also suggested a CNN-based motor performance prediction to aid the TO acceleration. In each study, cross-sectional image RGB values and rotor shape data of an interior permanent magnet motor are trained, and the computation time is reduced by up to 1/10 without quality loss. [Nakamura and Suzuki (2020)] proposed a CNN-based encoder and decoder model to optimize the structure for given design conditions without iteration. The proposed method used batch normalization in the encoder to increase the stability, and spatial adaptive denormalization in the decoder to reinforce the design area information. Using the proposed model, the computational time required to obtain an optimal material density distribution was reduced by 83% compared to the SIMP. [Kollmann et al. (2020)] developed a CNN-based DL model to predict the optimized metamaterial design for either maximizing bulk modulus, maximizing shear modulus, or minimizing Poisson's ratio. The data for the CNN model is generated by a TO framework, which involves an energy-based homogenization method and periodic boundary conditions. The proposed model was able to give faster but high-quality TO results even in low-end laptops. CNN-based DL models can provide faster TO algorithms for multiscale metamaterial systems. [Lee et al. (2020)] proposed a CNN-based recognition method for TO that eliminates the FEA step and accelerates the TO. CNN was used to train the topology of the images where regression problems were combined to calculate compliances and volume fractions for TO processes. The proposed method was used to solve TO problems, and implemented in CPU and GPU. GPU took comparably less time for training in CNN. [Abueidda et al. (2020)] developed a CNN model based on ResUnet for efficient 2D TO of materials with linear elastic and hyperelastic responses. The developed CNN model showed satisfactory robustness while achieving good agreement with negligible computation time in the results of an expensive computational nonlinear TO framework. [Wang et al. (2021b)] proposed a deep CNN-based TO with generalization ability. The architecture of the NN consists of encoding and decoding parts, and U-Net was used to improve performance. The performance of the proposed method was evaluated compared to SIMP for a typical optimization problem, showing that the computational cost can be significantly reduced without sacrificing much performance of the design solution. [Bielecki et al. (2021)] introduced a DL-based fine resolution structure generative TO. The proposed approach utilized the feedforward DNN and CNN. The novelty of this method was that the density and the nodal deflections are used for a wide range of design spaces. The 3D verification TO example showed one order of magnitude time saving compared with the traditional TO. [Qian and Ye (2021)] proposed dual-model ANN to accelerate gradient-based TO. The dual-model ANNs can greatly accelerate the design process because it serves as a surrogate model to replace the sensitivity calculation. The demonstrated 64×64 benchmark example was 137 times and 74 times more efficient in forward and sensitivity calculations, respectively, and held around 95% accuracy with only about 2,000 training points.

3.3 Efficiency improvement using other ML techniques

[Lei et al. (2019)] developed an ML-driven real-time TO paradigm under the moving morphable component-based solution framework. In that work, SVR-ML models were employed to establish an engineering intuition on optimized structures corresponding to various external loads. Since the design variables and the layout of the optimized structure were mapped in the ML model, parameters could be reduced, and real-time structure analysis was possible. The effectiveness of the proposed method was demonstrated by numerical examples. The use of KKT conditions may further increase the efficiency of the proposed methodology. Lynch et al. (2019)] introduced a GP-based framework that avoids the trial and error involved in the manual tuning of parameters for TO. This [framework consisted of two steps: a meta-learning step where a recommendation was drawn from similar problems and a metamodeling step where Bayesian optimization was used to efficiently optimize the parameters for the specific TO problem, in which the tuning parameters were read from a prior repository. The proposed method showed better efficiency than the line search in the simple demonstrated TO problems with one to three tuning parameters. [Kallioras et al. (2020)] proposed a two-phase methodology that relies on DL to accelerate the TO procedure. DBNs were integrated into SIMP to solve TO problems. DBN was sequentially connected to restricted Boltzmann machines and used to discover higher-order connections between the density values of each finite element of the domain along with the first iteration of the SIMP approach. The input and output data were the density fluctuation patterns of finite element discretization and the resulting distribution of density values provided in the initial step of SIMP, respectively. In various 2D and 3D benchmark TO problems, it was observed that the proposed method accelerated successfully regardless of parameters and datasets. [Strömberg (2020)] also used SVM as a geometric model for performing an efficient and detailed TO. These SVM-based hypersurfaces were used to set the design of nonlinear

FEA using Boolean and blending operations. A 3D design example with a plastic limit load constraint showed that the proposed metamodel-based design optimization can be implemented within a few minutes. [Jiang et al. (2020)] proposed a framework based on ML for moving morphable component-based TO parameter tuning by combining extra-trees-based image classifier integrated with a particle swarm optimization algorithm. This method avoids the need for manual parameters adjustment of methods of moving asymptote, the popular optimizer for TO, and two benchmark problems were used to evaluate its performance. Thus, said method saves a lot of manpower and obtains a comparably feasible structure in the design domain. [Zhang et al. (2021b)] proposed a simple NN-based TO via a neural reparameterization framework that can solve various TO problems. In the proposed method, reparameterization is performed to update the pseudo density, which is the design variable of the conventional TO, into an NN parameter. The sensitivity analysis is implemented with an automatic differentiation technique, and transfer learning is used to accelerate the proposed method. Numerical examples of various fields showed that the proposed method could stably obtain an optimized structure. [Gladstone et al. (2021)] presented a variational autoencoder (VAE)-based robust TO for structure design with the best average performance while reducing the response sensitivity to input uncertainties. In the proposed method, a fast solution search was possible because the NN surrogate model is used as the finite element solver, and the design is searched in the low-dimensional space of the VAE. The L-bracket and heat sink examples showed the effectiveness of the proposed approach. [Wang et al. (2022)] proposed a data-driven multiscale TO approach that enables multi-scale cellular designs through the selection of multiple microstructural classes. In this approach, latent variable GP enhanced with the sum of separable kernels is presented. In several fundamental frequency maximization problems for verification, multiclass designs using the proposed method always have better performances than single-class designs.

3.4 Efficiency improvement for multi-resolution approaches

For multi-resolution TO, multi-level data is used to efficiently calculate the near-optimal design. [Yu et al. (2019)] proposed a novel DL-based near-optimal topological design determination process. In the proposed method, low- and high-resolution images created under the same boundary conditions and optimization settings were used for training, and cGAN was connected to a CNN-based encoder to efficiently determine a near-optimal design of high-resolution. Numerical examples showed that the proposed method can determine the pixel values and compliance of the near-optimal structure with negligible computational cost. In addition, [Napier et al. (2020)] presented a new approach to achieving high-resolution TO designs by training ANNs on a set of optimizations divided into small patches. Test examples showed significant time savings because the optimal structure of the coarse mesh can be mapped to the refined mesh. [Keshavarzzadeh et al. (2021)] introduced a simulation parameter and image segment mapping approach based on a deep disjunctive normal shape model. Because high-resolution designs were generated using inexpensive low-resolution designs, the performances of various design candidates could be quickly explored. A 3D TO example showed that this framework provides designs close to the optimal design, and can be effectively used as initial guesses. [Wang et al. (2021a)] also proposed a training method to establish a mapping relationship between low-resolution and high-resolution structures. In that paper, the enhanced deep super-resolution NN and CNN were used to predict the optimized high-resolution structure, and the structure containing multiple geometric boundary conditions was predicted within a negligible time. [Behzadi and Ilies (2021)] introduced transfer learning based on a CNN that can handle a variety of high-resolution 3D design domains and enables real-time explorations according to changing design conditions. This method trains the source network using a low-resolution dataset and fine-tunes the target network using a much smaller high-resolution dataset. The real-time predictions achieved an average binary accuracy of 95% from multiple experiments. [Elingaard et al. (2022)] used CNN to parameterize the mapping from a set of coarse mesh parameters to a one-scale design of a fine mesh, thereby saving computational cost without solving the least square problem associated with the traditional de-homogenization approach. To train the NN, a loss function has been developed that guarantees a periodic output field along the local lamination orientations, and its robustness is enhanced by not using the underlying structural optimization problem. Numerical examples show that the proposed method is 5 to 10 times faster than the state-of-the-art de-homogenization method.

3.5 Accuracy improvement using ML techniques

Most studies have been conducted to increase the efficiency of TO using ML, but there are various studies in which ML is used to improve the accuracy of TO. [Cang et al. (2019)] presented a theory-driven mechanism to perform generative TO using the NNs. Deviations of the training data from the optimality conditions were quantified and used to learn new data points. This process is called theory-based ML and shows improved results in generating near-optimal solutions to TO over standard supervised learning models at the same computational cost. [Ates and Gorguluarslan (2021)] proposed a two-stage network model using convolutional

encoder-decoder networks. In the first stage, a DNN model was used in two parallel networks. Given a priori information in the first stage, the second stage was trained using binary cross-entropy to provide the final predictions. The proposed method used a new loss function binary cross-entropy to solve the errors caused by a structural disconnection in the existing studies. The 2D and 3D TO data sets for the verification of its generalization ability show that the two-stage network model improves the predictive ability and significantly reduces the compliance and volume fraction errors compared to the single network.

4 Machine learning for design synthesis

In new product design, the design space depends on expert opinion, the designer's creative skills, and the bounds of design variables, among other factors. In addition, the designer's expertise is required to perform design optimization, and a large amount of time and cost are required for product testing. Therefore, the design of a new product from similar design prototypes may lack diversity or have limitations in improving product performance. However, with the recent surge in computing power, various ML-based generative methods have been developed. If these cutting-edge computing technologies and ML frameworks are used well, design synthesis with diversity or improved performance can be performed without designer expertise using existing data.

The design synthesis tree structure is presented in Fig. 5. Depending on the purpose of design synthesis, Sect. 4 can be classified into 1) design diversity, 2) improvement of design quality and performance, and 3) human preference.

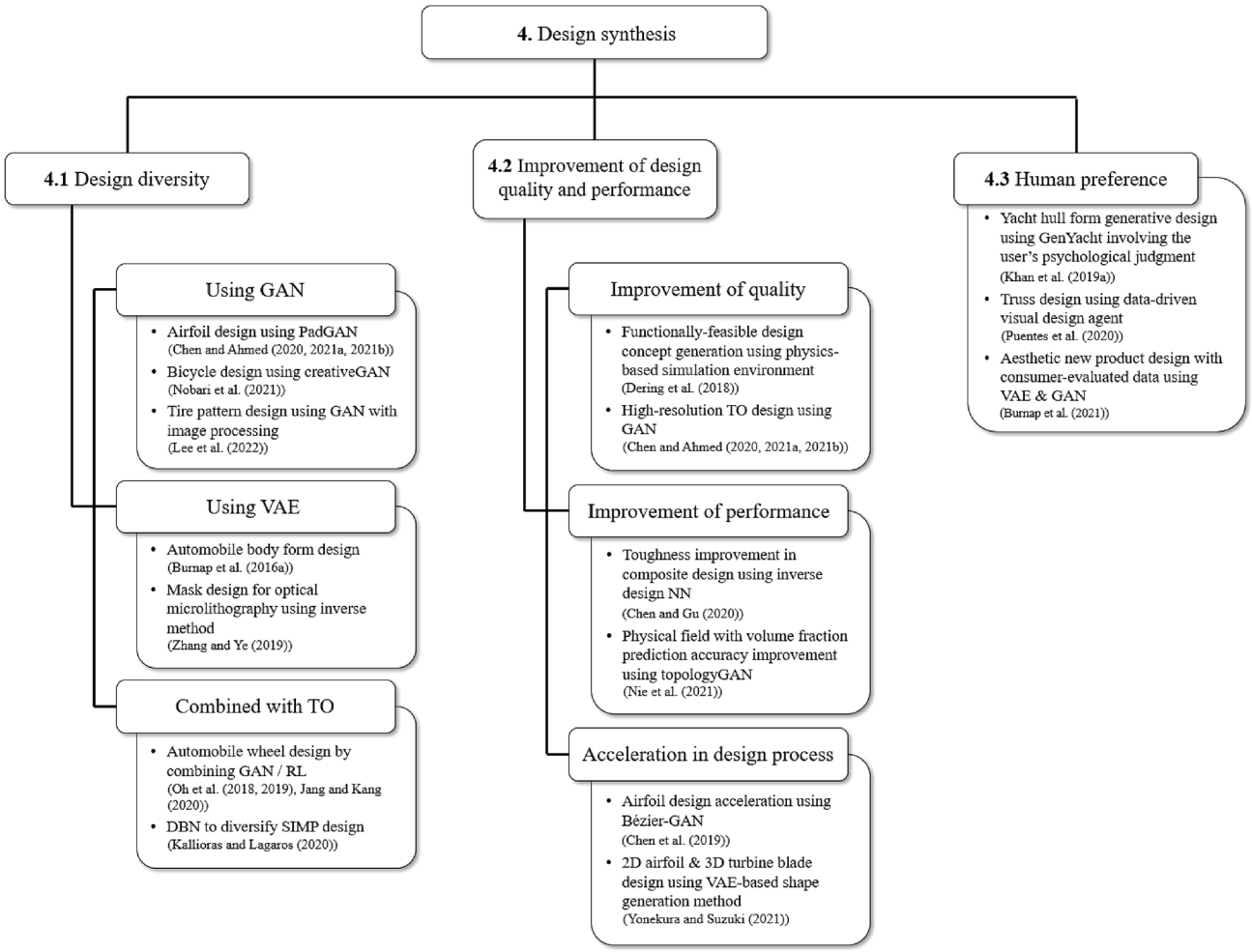


Fig. 5 Design synthesis tree structure

4.1 Design diversity

4.1.1 Design diversity using GAN and VAE

To increase the diversity of design synthesis, GAN and VAE are used among various ML techniques. While ML methods including NN can only function as a surrogate model, by using generative characteristics of GAN and VAE in the latent space, it is possible to obtain designs with diversity while satisfying the design conditions. [Chen and Ahmed (2020, 2021a, 2021b)] developed a new deterministic point process called performance augmented diverse GAN, which can generate designs with improved performances while covering the design space by measuring the performance of the generated model with diversity and quality scores. The real-world airfoil design used in the demonstration showed that the obtained result was a high-quality novel design. [Dai et al. (2019)] introduced a new design approach that can meet more individualized needs for efficient and custom manufacturing. In this approach, an architecture formulation using GAN was developed, and wristwatch wireframe design data are used for validation. [Nobari et al. (2021)] proposed an automated method, CreativeGAN, for generating novel designs. In that method, GAN was modified to identify the elements that uniquely induce a design and to create a design with those elements. The proposed method using a bicycle data set for validation showed that creative designs can be synthesized without human intervention and permit rethinking of design exploration. [Burnap et al. (2016a)] presented a new approach to estimating a new generative design within the design space using VAE. In that paper, 2D automobile body forms were used for demonstration, and designers can obtain new insights as new designs can be made according to body types and brands from 180,000 design data. [Zhang and Ye (2019)] proposed a DL-based inverse method to train constraints using VAE to automatically generate design candidates that satisfy the constraints. Inverse design of surface diffusion-induced morphology change and mask design for optical microlithography examples were used to demonstrate the performance of the method. [Lee et al. (2022)] proposed a deep generative tread pattern design framework to automatically generate various patterns that satisfy target tire performance. The proposed framework includes suitable image preprocessing, GAN, 2D image-based tire performance evaluation functions, design generation, design exploration and image post-processing methods. The numerical results generate various tire patterns that satisfy the requirements of target tire performance which varies with the seasons.

4.1.2 Design diversity combined with TO

To expand design exploration capabilities using ML in TO, a design automation process combining GANs and TO was proposed by [Oh et al. (2018, 2019)], and a generative design process based on RL was proposed by Jang and Kang (2020) along with a reward function that considers design diversity. Its efficiency was demonstrated by a case study of 2D automobile wheel designs. [Kallioras and Lagaros (2020)] presented DzAIN using DBN to diversify the design generation of SIMP. Through a 2D topology problem, it was verified that DzAIN successfully generates a prototype design. [Sun and Ma (2020)] developed a method of approaching density-based TO as a generative design formulation using RL-based algorithms. In particular, the ε-greedy policy and upper confidence bound method were used in this approach, and acceptable generation options were generated through an atmospheric diving suit problem, and 2D and 3D beam problems.

4.2 Improvement of design quality and performance

4.2.1 Design synthesis for quality and performance improvement

In this section, research on the quality and performance improvement of design synthesis is introduced. Most of the studies explained in Sect. 3 have improved the efficiency of TO algorithms by utilizing ML techniques or by proposing a surrogate model to replace TO iteration. Hence, the studies still require TO iterations even if the number of iterations is much reduced owing to ML. On the other hand, new design synthesis results are obtained without iterations by GAN trained with pre-training data, and the performance of these results is compared with the performance and accuracy of existing methods or designs. In addition, optimization in the latent space of GAN and AE makes it possible to accelerate design synthesis. To improve design quality, [Dering et al. (2018)] presented a physics-based simulation environment to alleviate the manual process problem of labeling training data used in DL training. In this environment, since the user can discover the correlation between the physical constraints related to the form of the generated design, it is possible to increase the possibility of creating a feasible design. [Li et al. (2019)] suggested a design methodology for multi-resolution TO where the proposed two-stage hierarchical refinement pipeline consisted of GAN for near-optimal prediction and super-resolution GAN for prediction at high-resolution. In the experiment for the heat transfer structure, it was demonstrated that the conductive heat transfer topology can be accurately estimated using negligible computational time. To enhance the design performance, [Chen and Gu (2020)]

proposed an inverse materials design method using generative inverse design networks which consist of two DNNs, namely predictor and designer, and the analytical gradient of the objective function is calculated using back propagation. The composite design problem was used for verification and showed that the toughness performance was higher than that of the gradient-based TO and genetic algorithm. [Shu et al. (2020)] also used GAN to generate 3D models. In this process, a newly generated design can be added to an existing training data set to generate high-performance models. Through the 3D aircraft model used in the demonstration, the process was repeated three times, resulting in improved geometric validity and performance. [Jiang and Fan (2019, 2020)] proposed conditional GLOnets, a global optimizer based on conditional NNs for electromagnetic device metasurface global optimization. In particular, this network can move design space to the high-performance region during the optimization process, and subsequent efficiency gradients were used for back propagation. The electromagnetic device design produced by the method agrees well with the best device produced by adjoint-based TO. [Yuan and Moghaddam (2020)] developed a design attribute GAN model to automatically generate fashion product images with desired visual attributes. In the performance evaluation results, the method is compared with the previous attribute GAN, and it is verified that attribute-aware image editing is performed with high accuracy. [Nie et al. (2021)] proposed a new methodology called TopologyGAN, in which the physical field calculated in the non-optimized domain is used as an input to the constructor of cGAN. TopologyGAN can reduce errors compared to cGAN, and a new U-SE-ResNet that combines the functions of U-Net and SE-ResNet was proposed to improve the overall accuracy. [Yamasaki et al. (2021)] proposed a data-driven topology design methodology. Since the elite material distribution was continuously updated and selected for training the dataset, various elite material distributions were generated through a deep generative model constructed with VAE. As this process was repeated, the performance of newly generated material distribution was further improved. The usefulness of the proposed method was verified through various 2D examples.

4.2.2 Design synthesis acceleration

[Chen et al. (2019)] utilized the Bézier-GAN to capture the low-dimensional latent space encoding the major shape variability of aerodynamic design. Through the airfoil design optimization problem, it was shown that optimization can be accelerated compared to other algorithms because the optimization was performed in the latent space. In addition, for electromagnetic metasurface optimization, [Shi et al. (2020b)] proposed a metasurface inverse design method consisting of CNN, AE, and optimized SVM. By entering the design target into the method, the metasurface structure can be obtained directly, and this reflects an improvement in the efficiency and acceleration of the design process. [Yonekura and Suzuki (2021)] proposed a conditional VAE-based shape generation method where the aerodynamic shapes are used as the input, and the performance goes into the continuous label. Then, specific shapes can be generated for specific performance latent vectors. For verification, 2D airfoil and 3D turbine blade design examples were introduced. In particular, in the turbine blade problem, more than 50,000 actual datasets were used, showing that different design trials are possible using the trained model.

4.3 Human preference

[Raina et al. (2019)] proposed a two-step framework that extracts information from historical human design strategies and allows design generation through observing design state sequences without additional information. [Puentes et al. (2020)] used the heuristic-guidance method to augment the decision-making of a data-driven visual design agent. The truss design problem showed that the agent can train and generate design strategies efficiently and superiorly in each method. [Khan et al. (2019a)] developed an interactive yacht hull form generative design process, GenYacht. The user can choose a design according to the hull design appearance and mechanical performance, and the process is repeated over and over until a satisfactory design is reached.

[Khan et al. (2019b)] presented a generative design approach that involves the psychology in the design exploration stage. The user's judgment was extracted as a psycho-physical distance metric, and sampling teaching–learning-based optimization was used to generate initial design alternatives. [Burnap et al. (2021)] utilized VAE and GAN for the aesthetic design of new products considering human judgement. Using consumer-evaluated data, the trained model predicts the aesthetics of the design well, and the newly generated design was aesthetically improved over the design used for training.

5 Applications and software tools

5.1 Applications

Recently, ML methods have been widely applied to facilitate design optimization or generate designs in various engineering applications. This section lists various ML methods and engineering applications. The engineering applications are divided into 1) structural and material design, 2) fluid, aerodynamics, or heat transfer, and 3) multidisciplinary and other applications as shown in Table 2, which also lists ML

Table 2 Engineering applications for ML

Methods	Structural and material design	Fluid, aerodynamics, and heat transfer	Multidisciplinary and other applications
ANN	FE model updating—Naranjo-Pérez et al. (2020), Floor rails of cab—Wang et al. (2021c), Optical flying head suspension—Lee et al. (2008), Dendritic growth—Liu and Wang (2019)	Re-entry vehicle—Parsonage and Maddock (2020), Design of a two-vane pump—Ma et al. (2020), Predict lift & surface pressure—Singh et al. (2017), Aerodynamic design—Sun and Wang (2019)	Single-stage rocket—Minisci and Vasile (2013), Unmanned space vehicle—Minisci and Vasile (2013), Nonlinear beam dynamics—Wang et al. (2019), Product form design—Diego-Mas and Alcaide-Marzal (2016), Concrete road bridges—García-Segura et al. (2017)
DNN, DBN, CNN, RNN, Bayesian inference	Crack damage detection—Cha et al. (2017), FE analysis of 2D structures—Jung et al. (2020), Structural reliability analysis—Gomes (2020), Mechanical plate against vibration—Li et al. (2020), Additive manufacturing—Bi et al. (2020), Metamaterial design—Kollmann et al. (2020),	Thermal conductor design—Li et al. (2020), Shape optimization—Zhang et al. (2021a), Aerodynamic robust optimization—Tao and Sun (2019), Heat source layout—Chen et al. (2020), Subsurface permeability—Yan and Zhou (2019), Multi-physics problems—Yao et al. (2020), Heat transfer TO—Lin et al. (2018, 2019)	Electromagnetic metasurface design—Shi et al. (2020b), Design of one-way slabs—Ferreiro-Cabello et al. (2018), Groundwater predictions—Müller et al. (2021), Shape optimization of airfoils—Lye et al. (2021), Design repository, additive manufacturing—Williams et al. (2019), Magnet motor—Doi et al. (2019), Sasaki and Igarashi (2019a, 2019b)
GAN, DCGAN, Conditional VAE	TO—Oh et al. (2018, 2019), Jang and Kang (2020), Design of airfoil—Chen and Ahmed (2020, 2021a, 2021b), 3D aircraft model—Shu et al. (2020), Microstructure design—Tan et al. (2020)	Conductive heat transfer topology—Li et al. (2019), Airfoil & turbine blade—Yonekura and Suzuki (2021)	Classifying the faults of an induction motor—Lee et al. (2017)
RL Deep, RL, GP, PNN, PINN	Seismic-resisting frame—Chhabra and Warn (2019), Structural reliability analysis—Jin (2020), Time-dependent reliability analysis—Li and Wang (2021), Metamaterial design—Bostanabad et al. (2019), Hydrogen storage—Patel and Choi (2012)	Inertial flow sculpting—Lee et al. (2019)	Shape optimization—Yonekura and Hattori (2019), Global routing—Liao et al. (2020), Photovoltaic and battery systems—Odonkor and Lewis (2019), Dielectric metasurfaces—Jiang and Fan (2019), Solution to PDEs—Raissi and Karniadakis (2018), Li and Mei (2021), McFall (2013), Solution to nonlinear PDEs—Raissi et al. (2019)
Decision tree, K-means, SVR, GIDNs, AutoML-GASOM, Probabilistic automata , Feature learning, Random forest	S-shaped side rail—Du et al. (2021), Automotive crashworthiness—Liu et al. (2015), FE analysis of 3D truss and 2D nonlinear multiscale problem—Capuano and Rimoli (2019), Composite materials—Chen and Gu (2020)	CFD analysis of relief valve—Shi et al. (2020a), Immiscible two-phase—Jabarullah Khan and Elsheikh (2019), Compression ignition engine—Owoyele et al. (2021), Design space exploration -Thole and Ramu (2020) Renewable energy mooring system—Pillai et al. (2019)	Primary flight control system—Garriga et al. (2019), Landing gear braking system—Garriga et al. (2019), Driving control of autonomous vehicles—An et al. (2020), Airfoil shape optimization—Tenne (2019), Aerodynamic flow—Zhang and Zhao (2021), Integrated voltage regulator—Trinchero et al. (2018), Wireless power transfer—Trinchero et al. (2018), Electromagnetic metasurface design—Jiang and Fan (2019, 2020), Customer choice predictions—Burnap et al. (2016b)

Table 3 ML/DL packages for structural and multidisciplinary optimization problems

Package/Library	Language	License	Link
pyOpt—Perez et al. (2012)	Python	LGPL	http://www.pyopt.org/
OpenMDAO—Gray et al. (2019)	Python	Apache license	https://openmdao.org/
OpenLSTO—Kambampati et al. (2018)	C++	Apache license	https://github.com/M2DOLab/OpenLSTO
Dymos—Falck et al. (2021)	Python	Apache license	https://github.com/OpenMDAO/dymos
Ārtap—Pánek et al. (2020)	Python	MIT	http://www.agros2d.org/artap/
DAFoam—He et al. (2020b)	C++	GPL-v3	https://dafoam.github.io/
WhatsOpt: a web application for multidisciplinary design analysis and optimization—Lafage et al. (2019)	Ruby, Javascript, HTML & Python	AGPL-3.0	https://github.com/OneraHub/WhatsOpt

methods grouped into five and explains why and how the ML methods are used in the applications.

5.2 Software tools

Several commercial software tools and open-source codes that have ML capabilities for use in design applications are available. ML/DL packages or libraries that are specifically for the structural and multidisciplinary optimization problems are provided in Table 3. A list of codes from works that are cited in this work is presented in Table 4. Table 5 provides a list of software that are essentially platforms on which data from different domains can be preprocessed and various ML models can be used to train the data for predictions and analysis. They are fundamentally ML as a service (MLaaS) platforms. All software is available on the cloud and some offer computations on the edge enabling technologies related to digital twins and advanced manufacturing. In addition, we also provide a list of packages within specific languages that contain different ML algorithms which permit training and predictions in Table 6.

6 Summary

In the past decade, ML technology has developed rapidly as computational performance has improved, and interest in topics using ML has also increased in structural and multidisciplinary optimization fields. However, the current literature lacks a thorough survey on structural and multidisciplinary optimization utilizing ML technology including optimization using ML to deal with uncertainty, TO, and design synthesis. This paper provides a comprehensive survey of ML algorithms mainly used in the field of structural and multidisciplinary optimization. From the survey, the following conclusions can be drawn and recommendations for future research are suggested.

- Although the NN model concept was first introduced in the 1940s, improvement in computing speed has made it possible to apply up-to-date ML techniques to structural and multidisciplinary optimization. As a result, better performances or new designs can be efficiently obtained.
- In general, if an ML/DL model is properly trained, it can infer results almost instantly over the domain of trained inputs, providing greatly accelerated design and analysis even when used with a laptop.
- Surrogate models are widely used to accelerate design optimization and are the most active research area among ML-based studies. By properly combining the data in the MF model condition, efficiency in surrogate modeling can be further improved. Significant time saving is possible when using multi-resolution data in TO.
- Combining IoT and FE simulations in the edge using PINN or other ML techniques for digital twin modeling permits the identification of failures long before they take place.
- In the design synthesis for new concept design, VAE and GAN are mainly used. The latent space can be utilized to increase design diversity or to obtain an efficiently optimized design. Many studies have been conducted on generating a new model architecture for design synthesis, and various attempts continue to be made.
- For repetitive computer simulations with minor changes in geometry, material properties, and loading conditions, employ ML techniques to model existing data as prior and be able to bypass the simulation itself for newer designs, thereby reducing the design time exponentially.
- As with several studies, providing source codes for benchmark examples can contribute to the development of ML methods in structural and multidisciplinary optimization. A more improved method than the existing methods can be proposed through comparison with the latest ML method.
- In ML-based surrogate modeling, there is insufficient information on guidelines regarding how to use the MF condition well or how many samples to use. In order to perform research on this, it is necessary to verify various

Table 4 Codes available from the papers reviewed in this work

Topic	Language	Link
Iterative surrogate model optimization (ISMO)—Lye et al. (2021)	Python	https://github.com/kjetil-lye/iterative_surrogate_optimization
GP using CPKL tree-GEP—Jin (2020)	MATLAB	https://github.com/seungsab/CPKL_using_Tree-GEP
GP Bayesian estimation in ordinary differential equations—Barber and Wang (2014)	MATLAB	https://github.com/odegp/code
Finite element machine classifier (FEMa)—Pereira et al. (2020)	C	https://github.com/danillorp/libFemClassifier
PINN—Raissi et al. (2019)	Python	https://github.com/maziarraissi/PINNs
Hidden physics models (PINN)—Raissi and Karniadakis (2018)	MATLAB	https://github.com/maziarraissi/HPM
Interactive self-organizing maps (iSOM)—Thole and Ramu (2020)	MATLAB	https://ed.iitm.ac.in/~palramu/SOM/
JAX-CFD: CFD in JAX—Kochkov et al. (2021)	Python	https://github.com/google/jax-cfd
Conditional deep surrogate models—Yang and Perdikaris (2019)	Python	https://github.com/PredictiveIntelligenceLab/CADGMs
MF Bayesian optimization via deep NN—Li et al. (2020)	MATLAB	https://github.com/kirthevasank/mf-gp-ucb) https://github.com/YehongZ/MixedTypeBO https://github.com/zi-w/Max-value-Entropy-Search/
MF physics-constrained NN and its application in materials modeling—Liu and Wang (2019)	Python	https://github.com/tensorflow/tensorflow
A composite NN that learns from MF data—Meng and Karniadakis (2020)	Python MATLAB	https://github.com/lululxvi/deepxde https://github.com/pratikkakkar/deep-diff
Transfer learning-based MF physics-informed DNN—Chakraborty (2021)	MATLAB	https://colab.research.google.com/notebooks/tensorflow_version.ipynb
Uncertainty modeling and runtime verification for autonomous vehicles driving control—An et al. (2020)	MATLAB	https://github.com/DongdongAn/DrivingStyleClassification
Learning DNN surrogate models for high-dimensional uncertainty quantification—Tripathy and Bilionis (2018)	Python	https://github.com/rohitkt10/deep-uq-paper
Deeply uncertain: Comparing methods of uncertainty quantification in DL algorithms—Caldeira and Nord (2020)	Python	https://github.com/deepskies/DeeplyUncertain-Public
Self-direct online ML for TO—Deng et al. (2020)	MATLAB	https://github.com/deng-cy/deep_learning_topology_opt
TOuNN: TO using NNs—Chandrasekhar and Suresh (2021)	Python	https://www.ersl.wisc.edu/software/TOuNN.zip
NNs for TO—Sosnovik and Oseledets (2019)	Python	https://github.com/ISosnovik/top
Multi-stage DNN accelerated TO—Bielecki et al. (2021)	MATLAB	https://github.com/dustin-bielecki/Corner-Based-Topology-Optimization-Dataset
Gradient-based TO design with dual-model ANNs—Qian and Ye (2021)	Python	https://github.com/hkust-ye/cqian_dual-model_neural_network/tree/master
MMC-based TO—Jiang et al. (2020)	MATLAB	https://github.com/yoton12138
MO-PaDGAN: Reparameterizing engineering designs for augmented multi-objective optimization of a real-world airfoil design example—Chen and Ahmed (2021a)	Python	https://github.com/wchen459/MO-PaDGAN-Optimization
PaDGAN: A GAN for performance augmented diverse designs—Chen and Ahmed (2021b)	Python	https://github.com/wchen459/PaDGAN
DL-based inverse method for layout design—Zhang and Ye (2019)	Python	https://github.com/yzhangbx120/Inverse-Design
Generative design by using exploration approaches of RL—Sun and Ma (2020)	MATLAB	https://github.com/Nick0095/STO_exploration https://www.youtube.com/watch?v=EbB2id-1n2U
Generative NNs for the inverse design of metasurfaces—Jiang and Fan (2020)	Python	https://github.com/jonfanlab/GLOnet
TopologyGAN: TO using GANs-based on physical fields over the initial domain—Nie et al. (2021)	Python	https://github.com/zhenguonie/2020_TopologyGAN
Aerodynamic design optimization and shape exploration using GANs—Chen et al. (2019)	Python	https://github.com/IDEALLab/airfoil-opt-gan

Table 5 ML as a service (MLaaS) platforms

Software	Link
AWS AI	https://aws.amazon.com/tr/machine-learning/
Azure Machine Learning	https://azure.microsoft.com/services/machine-learning/
Cnvrg.io	https://cnvrg.io/
CoCalc	https://cocalc.com/
Google Colab	https://colab.research.google.com/
H2O AI	https://www.h2o.ai/
IBM Cloud	https://www.ibm.com/cloud/learn/machine-learning
KNIME	https://www.knime.com/
MyDataModels	https://www.mydatamodels.com/
Neural Designer	https://www.neuraldesigner.com/
Spell	https://spell.ml/
Vertex AI	https://cloud.google.com/vertex-ai
Weka	https://www.cs.waikato.ac.nz/ml/weka/

Table 6 Software packages with ML algorithms

Package	Language	License	Link
Accord.Net	C#	LGPLv3	https://github.com/accord-net/framework/wiki/Getting-started
Caffe	C++	BSD	https://caffe.berkeleyvision.org/
Dlib	Python / C++	Boost	http://dlib.net/
Keras	Python	MIT	https://keras.io/
Mahout	Java	Apache	https://mahout.apache.org/
NumPy	Python / C	License	https://numpy.org/
OpenNN	C++	BSD	https://www.opennn.net/
Orange3	Python / C++/	LGPL	https://orangedatamining.com/
PyTorch	C	GPLv3	https://pytorch.org/
Scikit-learn	Python / C++	BSD	https://scikit-learn.org/stable/
Scipy	Python / C++	BSD	https://scipy.org/
Shogun	Python	BSD	https://www.shogun-toolbox.org/
Singa	C++	BSD3	http://singa.apache.org/
Spark MLlib	C++/ Python / Java	Apache License	https://spark.apache.org/mllib/
TensorFlow	Java / Scala / Python / R	Apache License	https://www.tensorflow.org/
TensorFlow.js	Python / Go / Java / C / R	Apache License	https://www.tensorflow.org/js?hl=en
	Java	Apache License	

characteristics of examples and engineering problems, and additional research will need to be performed.

– When there is insufficient data when constructing a DNN, there is a limitation for optimization. Insufficient data for a model requiring a large number of hyperparameters leads to overfitting and poor prediction accuracy. Obviously, there will be areas in which it is difficult to generate a large number of data in structural and multidisciplinary optimization. Transfer learning to solve this problem and meta-learning methods to determine the initial model need to be studied further.

Appendix. ML methods widely used in the context of structural and multidisciplinary optimization

ML algorithms can be categorized into four groups: 1) classification, 2) regression, 3) clustering, and 4) dimension reduction as shown in Fig. 6. Classification and regression are both supervised learning algorithms, where the main idea is to generate a prediction model. If the predicted response is discrete, it is a classification problem, whereas if the response is continuous, then it is a regression problem. Therefore, in general, the ML algorithms used for classification and regression are very similar. The

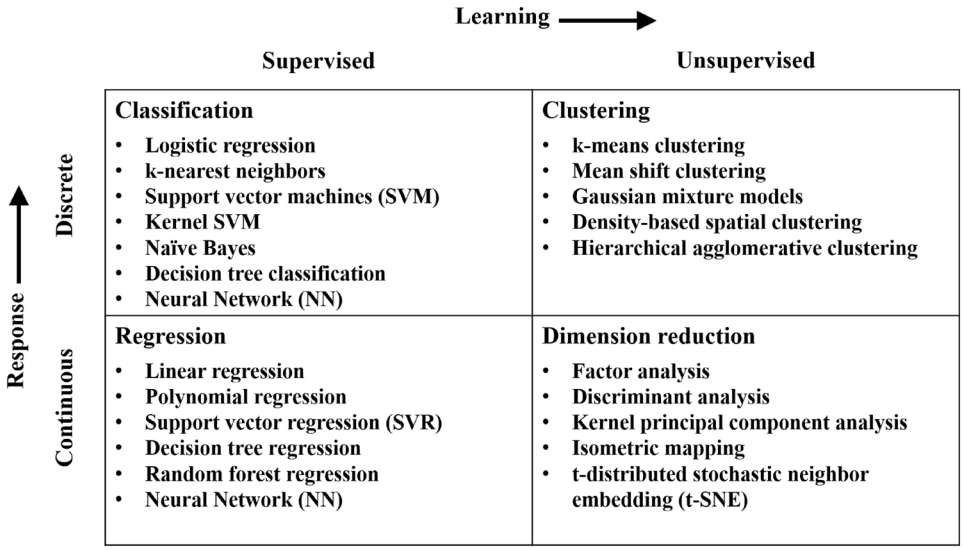


Fig. 6 Categories of ML problems

most commonly used classical ML algorithms for classification problems include logistic regression [Cox (1958)], k-nearest neighbors [Fix and Hodges (1989)], support vector machines (SVM) [Cortes and Vapnik (1995)], kernel SVM, naive Bayes, decision tree classification, and random forest classification. The most commonly used classical ML algorithms for regression problems include simple linear regression, multiple linear regression, polynomial regression, Kriging, support vector regression (SVR), decision tree regression, and random forest regression.

Clustering is similar to classification in that they are both used for grouping the data. The main difference is that classification is used to categorize labeled data, whereas clustering detects patterns within an unlabeled data set. Therefore, the classification is a supervised learning algorithm, whereas the clustering is an unsupervised learning algorithm. The most commonly used classical ML algorithms for clustering problems include k-means, mean shift clustering, Gaussian mixture models, density-based spatial clustering, and hierarchical agglomerative clustering.

Dimension reduction aims to reduce the number of input variables in a dataset, thereby protecting against the curse of dimensionality, which makes the algorithm difficult to run as the dimensions of the data increase. Data from a large dimensional space is transformed into a smaller dimensional space ensuring that it provides similar information. Dimension reduction methods can be further categorized into linear methods and non-linear methods. The most commonly used linear learning algorithms for dimension reduction include principal component analysis [Wiener (1938)], factor analysis [Harman (1976)], linear discriminant analysis [Fisher (1936)], and singular value decomposition [Golub and Reinsch (1971)]. The nonlinear algorithms include kernel principal component analysis, isometric mapping, and t-distributed stochastic neighbor embedding (t-SNE). Among the ML methods listed in Fig. 6, we briefly explain ML methods that are widely used in the context of structural and multidisciplinary optimization in the following subsections.

A.1 Linear regression

Linear regression [Montgomery et al. (2021)] models the relationship between the response variable (dependent) y and one or more independent variables x. If there exists only one independent variable, then it is called simple linear regression. The fundamental idea in linear regression is to find the coefficients of the basis functions that best model the data. Ordinary least squares (OLS) are the most common method used to train the model with the given data to estimate the unknown coefficients. Function nonlinearity is modeled using complex basis functions while keeping the regression linear.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n + \varepsilon = x^{fT}\boldsymbol{\beta} + \varepsilon \tag{A1}$$

where β_0 and β_n are the unknown coefficients and ε is the error term.

A.2 Gaussian process (GP)

GP [Rasmussen (2003)], also known as Kriging when the mean of GP is zero, is a stochastic approach that finds wide use in regression, classification, and unsupervised learning. It is usually utilized in the linear regression framework while using the Gaussian kernel as the basis function. It is the preferred approach for inference on functions as well. GP is a generalization of Gaussian probability distribution in which every finite collection of random variables has a multivariate Gaussian distribution. GP is a distribution over functions with a continuous domain such as time or space.

Since GP provides model prediction as well as prediction error estimates, even when the simulation is deterministic, it is sought after to be used as surrogates in design and analysis of expensive computer experiments. Since GP metamodels can fit complicated surfaces well, it is suited for fitting accurate global metamodels. A GP is completely specified by mean function $m(\boldsymbol{x})$ and covariance function $k(\mathbf{x}, \mathbf{x}')$ as

$$f(\mathbf{x}) \sim GP\big(m(\mathbf{x}),\, k(\mathbf{x}, \mathbf{x}')\big) \tag{A2}$$

GP can be extended to multiple outputs by using multiple means and covariances. It permits easy interpolation of data and has an inbuilt mechanism to account for noise. Furthermore, GP can quantify the uncertainty about the prediction and have conditional distributions that allow adaptive sampling or Bayesian studies. Owing to the fact that GP models are regularly evaluated on a grid leading to multivariate normal distributions and the computational time for calculating the inversion and determinant of $n \times n$ covariance matrix is of O^3, using GP is a challenge while using large scale datasets. Recently, approaches such as matrix–vector multiplication [Gardner et al. (2018a), (2018b)], [Dong et al. (2017)] and sparse GP [Cutajar et al. (2016)] have been developed to reduce the amount of computation when the data set is more than 100 k.

A.3 Artificial neural network (ANN)

In the 1940s, [McCulloch and Pitts (1943)] formulated the first NN model. Since its inception, NN has found interest among both researchers and applications in various domains. As a result, better algorithms and more powerful networks have been developed. ANN refers to a biologically inspired sub-domain of artificial intelligence (AI) modeled based on the network of the brain. Akin to the human brain, ANNs have neurons (called nodes) which are connected to each other in different layers of the networks as shown in Fig. 7. The basic idea of ANN is that an input vector **x** is weighted by **w** and along with bias **b**, subjected to an activation function f that is linear or nonlinear to produce the output y as given as

$$y = f\big(\mathbf{w}^{\mathrm{T}}\mathbf{x} + \mathbf{b}\big) \tag{A3}$$

The weights in Eq. (A3) are optimized during training until a specified level of accuracy is reached by the network. Based on the application, there are many activation functions used in ANN, namely sigmoid, hyperbolic tangent, rectifier linear unit (ReLU), Heaviside, signum, and softmax functions [Karlik and Olgac (2011)]. Researchers have also developed application specific activation functions (Wuraola and Patel 2018, [Gomes and Ludermir (2013). Since ANN deals with multidimensional data, approaches such as StandardScaler, RobustScaler, MinMaxScaler, and Normalizer for data scaling, can be used for data processing and can prevent convergence to zero or diverge to infinity during the learning process.

ANN is broadly classified into two categories such as feed-forward NN and feed backward NN. In the feed-forward NN, the information will pass only in the forward direction i.e., from the input layer to the hidden layer (if any) and then to the output. Single-layer perceptron, multi-layer perceptron, and radial basis function networks are examples of feed-forward NN. In the feed backward NN, the inputs are fed in the forward direction and errors are computed to be propagated in the reverse (hence the terminology back) direction to the previous layers, so as to reduce the error in the cost function by readjusting the weights. Examples include Bayesian regularised NN and Kohonen's self-organizing map. The loss function is computed as the difference between the prediction and the target after each feedforward pass. In the backpropagation process, the optimizer trains parameters such as weights and biases iteratively through optimization to minimize the loss function.

ANNs can be used for both regression and classification problems which are techniques in predictive modeling. In the context of classification, since ANN works by splitting the problem into layered networks of simpler elements, ANNs are reliable when the tasks involve many features. The most attractive feature of ANNs is that they provide predictive capability by mapping any number of inputs and outputs. Upon training, the predictions are fast and cheap.

NNs are typically black box approaches. That is, one might not be able to capture the influence of independent variables on dependent variables. Overfitting is a fundamental challenge of ANN as it depends predominantly on training data. With traditional CPUs, ANNs were expensive in terms of computational time to train the network, but the invention of cloud computing and increased computing power have relieved the computational burden. However, researchers began to focus on more complex problems and

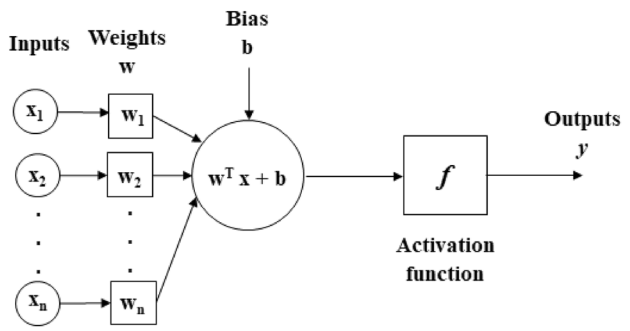


Fig. 7 Simple architecture of ANN

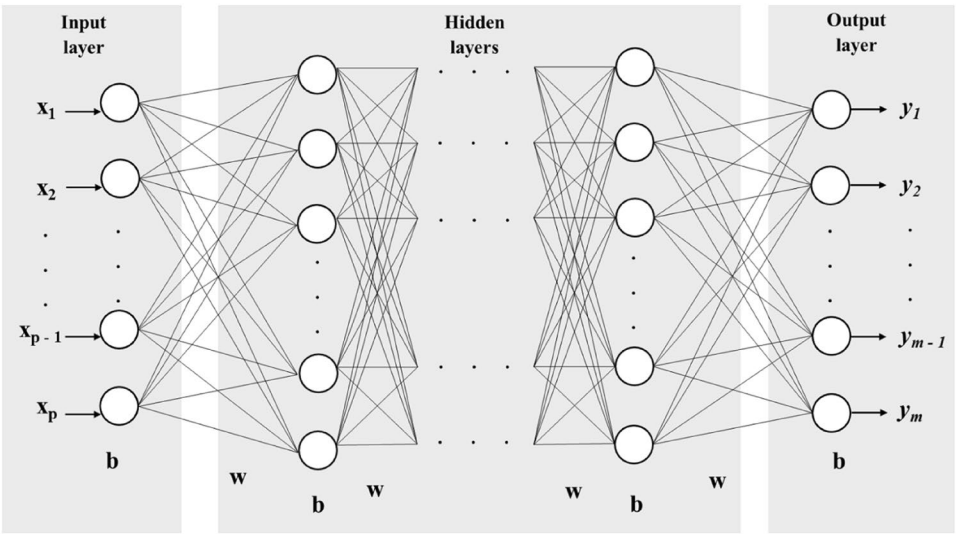


Fig. 8 Architecture of DNN

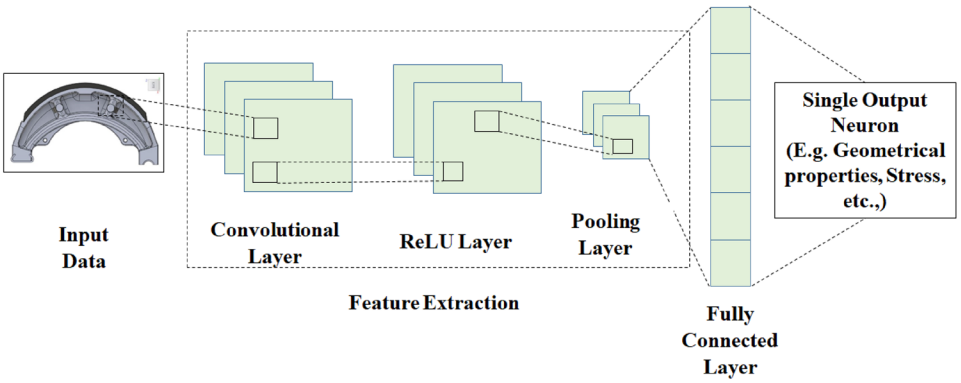


Fig. 9 Architecture of CNN

used more layers to train on large sets of data, resulting in longer computational times with multiple training iterations.

A.4 Deep neural network (DNN)

DNN is created when NNs are stacked one after the other. The primary difference between the conventional NN and DNN is that the former has one or two hidden layers and the latter has several hidden layers as shown in Fig. 8. Each circle in the figure calculates a weighted sum of the input vectors and bias following which a nonlinear function is applied to obtain the output. DNNs can handle functions with limited regularity and are powerful for high-dimension problems. The basic idea of DNN is to approximate a function with a non-linear activation function [Emmert-Streib et al. (2020)], with n hidden layers as represented in

$$\mathbf{DNN}(y) = \mathbf{w}^{(n)}\mathbf{x}^{(n)} + \mathbf{b}^{(n)} \tag{A4}$$

and

$$\mathbf{x}^{(k+1)} = \sigma\left(\mathbf{w}^{(k)}\mathbf{x}^{(k)} + \mathbf{b}^{(k)}\right), \quad k = 0, 1, \cdots, n-1 \tag{A5}$$

where $\mathbf{w}$ and $\mathbf{b}$ are the weights and biases of the network and σ is the activation function. DNN is more complex in connecting layers than a network with 1 or 2 hidden layers and has the automatic feature extraction capability. Therefore, when larger training data is used, the DNN can provide accurate predictions compared to classical ML algorithms where the accuracy is kept fairly constant.

There are three major classes of DNNs, namely supervised, semi-supervised, and unsupervised DNNs. Examples of supervised learning algorithms include deep feed-forward networks (DFNNs) and CNNs. Restricted Boltzmann machines, autoencoders (AEs), GANs, and long short-term memory networks (LSTMs) are examples of unsupervised learning algorithms. Recurrent neural networks (RNN) are an example of semi-supervised learning techniques.

While solving complex problems such as image classification, natural language processing, and speech recognition, DNN is more useful than shallow networks. DNNs typically outperform other approaches when the data is large. DNN architectures are very flexible to adapt to new problems and can work with any data type. Getting trapped in the local

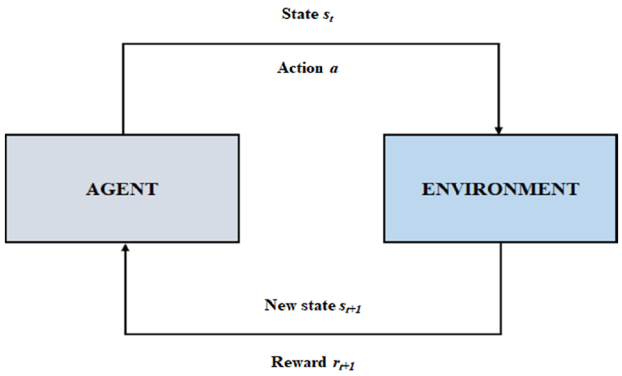


Fig. 10 Basic setting of RL

minima, vanishing gradient, and overfitting are some of the challenges associated with DNN training that require large data.

A.5 Convolutional neural network (CNN)

One of the most widely used DNNs are the CNNs [Fukushima (1988)]. While ANN is inspired by the human brain, CNNs are inspired by the human optical system and are predominantly applied to imaging analysis. CNNs consist of two operations, namely convolution and pooling. Unlike ANNs, in CNNs the neurons in one layer are connected to nearby neurons in the next layer. This leads to a significant reduction in the number of parameters in the network. A typical CNN consists of an input, an output, and multiple hidden layers which consist of a series of convolutional layers (filters or convolution kernels) as shown in Fig. 9. ReLU is the typical activation function used, followed by operations such as pooling layers, fully connected layers, and normalization layers. Backpropagation is used for error minimization and weight adjustment. [Wu (2017)] provides a tutorial on CNN. Compared to CPU-based architectures, CNNs with GPU-based architectures take less time for training, because GPU vastly is superior in the computation of dense algebraic kernels, such as matrix–vector multiplication, in which DL algorithms are mainly composed.

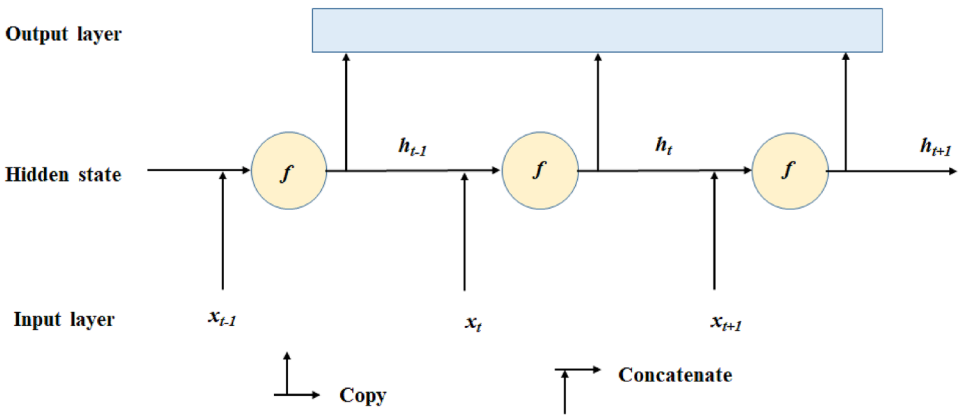
CNNs can easily process high-dimensional inputs such as images. CNNs are good at extracting local information from the text and exploring meaningful semantic and syntactic meanings between phrases and words. Also, the natural composition of text data can be easily handled by a CNN's architecture. CNNs need large data for training and hence are computationally intensive. Encoding the position and orientation of objects is still a challenge in CNN.

A.6 Reinforcement learning (RL)

RL [Sutton and Barto (2018)] is one of the paradigms of ML algorithms where the agents learn by interacting with the environment. RL works on trial and error-based learning and maximizes the reward rather than finding the hidden structure unlike other ML algorithms. As can be seen in Fig. 10, an RL agent performs an action 'a' while transiting from a state s_t to s_{t+1} and is rewarded r_{t+1} for the action a_t and this process is repeated iteratively to maximize the reward. The probability of transition to the new state is expressed by $P(s_{t+1} \mid s_t, a_t)$. The best sequence of actions that an RL agent can make is called a policy and the entire set of actions from start to finishthat an agent performs is called an episode. Usually, the dynamics of the RL problem can be captured by using a Markov decision process.

RL usually performs better in solving complex problems compared to other standard learning techniques. If no training data set is available, it is bound to learn from experience. RL focuses on achieving long-term results that are difficult to accomplish by other techniques. Similar to other ML techniques, RL requires large data and is computationally expensive. RL can be heavily affected due to the curse of dimensionality. When the conventional RL is combined with DL, deep RL can be set up. The deep RL uses DNNs to calculate rewards, and policies that are usually accomplished by a state of action pairs in RL. The deep RL can be

Fig. 11 RNN with hidden memory state



employed where there exists a complex state and very high computations are required (Fig. 10).

A.7 Recurrent neural network (RNN)

RNN [Rumelhart et al 1986] is one of the common semi-supervised learning algorithms that use sequential data (or ordered data) for training. Examples of ordered data are DNA sequence, financial data, and time-series data. RNN uses the current input as well as the past history of inputs that it has learned through the hidden state while making decisions. Typically RNNs consists of an input layer, a hidden layer, and an output layer as shown in Fig. 11. The number of neurons in the hidden layer of RNNs should be between the number of inputs and the number of outputs. The key feature of RNN is that it makes a copy of the output and sends it back into the network. Thus, the past information gets stored. The change in the knowledge of the network is updated in the hidden state at every time step and the update can be expressed as

$$h_t = f_w\left(x_t,\ h_{t-1}\right) \tag{A6}$$

where h_t is the new hidden state, $h_{t\text{-}1}$ is the past hidden state, x_t is the current input, and f_w is the fixed function with trainable weights.

These algorithms commonly find application in ordinal or temporal problems such as image captioning, speech recognition, and natural language processing. Similar to spatial data being efficiently processed by CNNs, RNNs are designed to process the sequential data in an efficient manner. As the number of time steps increases, the number of model parameters in the RNN model does not increase. While training an RNN, error gradients are used to update the network weights. Sometimes the error gradients can accumulate resulting in large updates of weights (exploding gradients) and an unstable network. On the other hand, if the weight updates are small, one faces the problem of vanishing gradients. These are the two major issues associated with the RNNs. In order to solve the gradient problem, weight initialization methods such as Xavier initialization and He initialization, gradient clipping, and batch normalization are used, or an LSTM or GRU is devised. When timely dependencies in sequences need to be captured, RNN are one of the best choices. However, recent developments such as Transformers [Vaswani et al, (2017)] can outperform RNN in such applications.

A.8 Variational autoencoder (VAE)

An autoencoder (AE) is a type of unsupervised learning that learns unlabeled data and has traditionally been used for dimensionality reduction and feature learning, but recently it has gained a lot of popularity as a generative model that can generate data similar to training data. Since VAE (Kingma and Welling 2013) is based on an AE, it consists of two parts: encoder and decoder. However, unlike AE, which represents a latent vector as a value, the latent vector of VAE uses a density function. Since VAE is based on a probabilistic model, it has computational flexibility. This latent vector is used to predict an input image, and VAE training is performed with the goal of reducing the difference between the generated image and the input image as shown in Fig. 12. Finally, evidence lower bound and re-parameterization tricks are used to perform optimization. The main advantage of VAE is that it is useful to perform other tasks such as design optimization in the latent space using the latent vector information. However, since the density is not obtained directly, the quality of the generated model may be somewhat inferior to the direct density methods such as pixelRNN or pixelCNN, and the generated image is relatively blurry compared to GAN.

A.9 Generative adversarial network (GAN)

GAN [Goodfellow et al. (2014)] as shown in Fig. 13 trains a model that samples a latent vector from a simple distribution and generates it as an image based on the game-theoretic approach. The objective function of GAN consists of a discriminator output for real data, and a discriminator output for generated fake data. Because the generator and discriminator train with the goal of minimizing and maximizing the objective function, respectively, GAN is called a minmax

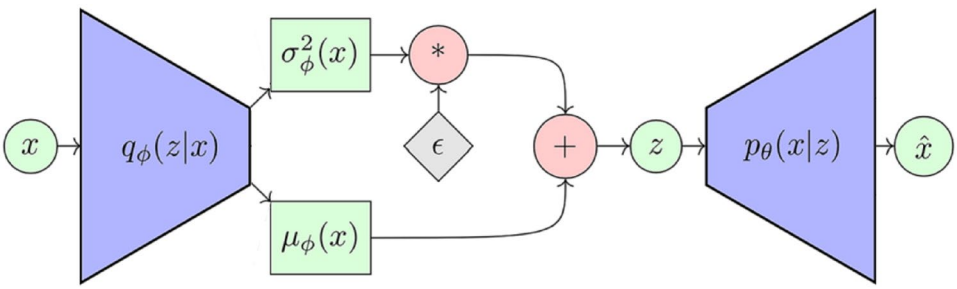


Fig. 12 Architecture of VAE (Asperti et al. 2021)

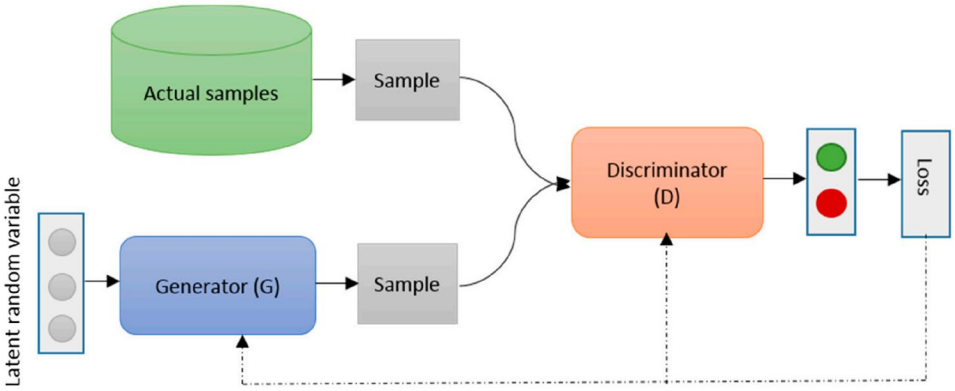


Fig. 13 Architecture of GAN (Alom et al. 2019)

game. The story of a counterfeiter (generator) and a police officer (discriminator) is an easy-to-understand example of the concept of GAN. When a counterfeiter creates a counterfeit currency, the police can determine whether it is genuine or not, and in the process, the generator and discriminator evolve competitively to generate a more authentic counterfeit currency.

GAN is difficult to apply to various fields due to unstable learning ability; consequently, a DCGAN [Radford et al. (2015)] with CNN in the generator part was developed. In addition to the GAN model, various models such as conditional GAN (cGAN), boundary equilibrium GAN, and super-resolution GAN have been developed to improve performance and for application to new fields. The main advantage of GANs is that it is possible to create new and novel images.

A.10 Ensemble methods

Ensemble methods are one paradigm of ML techniques that have become popular during the past three decades [Bishop (1995)], where several learning algorithms are used to train and solve a problem. Boosting, bagging [Bühlmann (2012)], and stacking (Džeroski and Ženko 2004) are the most widely used approaches in ensemble methods. AdaBoost [Rätsch et al. (2001)], gradient boosting (Friedman 2001), extreme gradient boosting [Chen and Guestrin (2016)], and light gradient boosting [Ke et al. (2017)] are a few algorithms that are more frequently used in boosting. Bagging meta-estimator and random forest are the popular ensemble algorithms in bagging.

Ensemble methods are used to improve the accuracy of the model by reducing the variance. Bagging is a variance reduction technique whereas boosting and stacking's objective is to reduce the bias and not the variance. Ensembles have been shown to serve as insurance against bad predictions and issue a red flag when one of the models is performing inconsistently on a consistent basis, especially at regions of interest. Eventually, all the ensemble algorithms attempt to improve the model accuracy. The generalization ability of a single learner is not as good as ensemble methods, since it uses multiple learners, and this is one of the major advantages of using ensemble methods.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Replication of results In this review paper, we do not provide any results to replicate.

References

Abueidda DW, Koric S, Sobh NA (2020) Topology optimization of 2D structures with nonlinearities using deep learning. Comput Struct 237:106283

Abueidda DW, Lu Q, Koric S (2021) Meshless physics-informed deep learning method for three-dimensional solid mechanics. Int J Numer Meth Eng 122(23):7182–7201

Acar E, Rais-Rohani M (2009) Ensemble of metamodels with optimized weight factors. Struct Multidisc Optim 37(3):279–294

Acar E, Solanki K (2009) System reliability based vehicle design for crashworthiness and effects of various uncertainty reduction measures. Struct Multidisc Optim 39(3):311–325

Adeli H, Park HS (1995) A neural dynamics model for structural optimization—theory. Comput Struct 57(3):383–390

Alom MZ, Taha TM, Yakopcic C, Westberg S, Sidike P, Nasrin MS, Asari VK (2019) A state-of-the-art survey on deep learning theory and architectures. Electronics 8(3):292

Amsallem D, Farhat C (2008) Interpolation method for adapting reduced-order models and application to aeroelasticity. AIAA J 46(7):1803–1813

An D, Liu J, Zhang M, Chen X, Chen M, Sun H (2020) Uncertainty modeling and runtime verification for autonomous vehicles driving control: A machine learning-based approach. J Syst Softw 167:110617

Asperti A, Evangelista D, Piccolomini EL (2021) A survey on variational autoencoders from a green AI perspective. SN Computer Science 2(4):1–23

Ates GC, Gorguluarslan RM (2021) Two-stage convolutional encoder-decoder network to improve the performance and reliability of deep learning models for topology optimization. Struct Multidisc Optim 63(4):1927–1950

Banga S, Gehani H, Bhilare S, Patel S, Kara L (2018). 3d topology optimization using convolutional neural networks. arXiv preprint arXiv:1808.07440.

Baraldi P, Mangili F, Zio E (2015) A prognostics approach to nuclear component degradation modeling based on Gaussian process regression. Prog Nucl Energy 78:141–154

Barber D, Wang Y (2014). Gaussian processes for Bayesian estimation in ordinary differential equations. In *International conference on machine learning* (pp. 1485–1493). PMLR.

Bataleblu AA (2019) Computational intelligence and its applications in uncertainty-based design optimization. In *Bridge Optimization-Inspection and Condition Monitoring*. IntechOpen.

Behzadi MM, Ilieş HT (2021) Real-time topology optimization in 3D via deep transfer learning. Comput Aided Des 135:103014

Bendsøe MP (1989) Optimal shape design as a material distribution problem. Struct Optim 1(4):193–202

Bi S, Zhang J, Zhang G (2020) Scalable deep-learning-accelerated topology optimization for additively manufactured materials. arXiv preprint arXiv:2011.14177.

Bielecki D, Patel D, Rai R, Dargush GF (2021) Multi-stage deep neural network accelerated topology optimization. Struct Multidisc Optim 64(6):3473–3487

Bishop CM (1995) Neural networks for pattern recognition. Oxford University Press, UK

Bostanabad R, Chan YC, Wang L, Zhu P, Chen W (2019) Globally approximate gaussian processes for big data with application to data-driven metamaterials design. J Mech Des 141(11):111402

Bühlmann P (2012). Bagging, boosting and ensemble methods. In Handbook of computational statistics (pp. 985–1022). Springer, Berlin, Heidelberg.

Burnap A, Pan Y, Liu Y, Ren Y, Lee H, Gonzalez R, Papalambros PY (2016b) Improving design preference prediction accuracy using feature learning. J Mech Des 138(7):071404

Burnap A, Liu Y, Pan Y, Lee H, Gonzalez R, Papalambros PY (2016a) Estimating and exploring the product form design space using deep generative models. In: International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (Vol. 50107, p. V02AT03A013). American Society of Mechanical Engineers.

Burnap A, Hauser JR, Timoshenko A (2021) Design and evaluation of product aesthetics: A human-machine hybrid approach. Available at SSRN 3421771.

Caldeira J, Nord B (2020) Deeply uncertain: Comparing methods of uncertainty quantification in deep learning algorithms. Mach Learn: Sci Technol 2(1):015002

Cang R, Yao H, Ren Y (2019) One-shot generation of near-optimal topology through theory-driven machine learning. Comput-Aided Des 109:12–21

Capuano G, Rimoli JJ (2019) Smart finite elements: a novel machine learning application. Comput Methods Appl Mech Eng 345:363–381

Cerbone G (1992) Machine learning techniques in optimal design. In: Artificial Intelligence in Design'92 (pp. 699–717). Springer, Dordrecht

Cha YJ, Choi W, Büyüköztürk O (2017) Deep learning-based crack damage detection using convolutional neural networks. Comput-Aided Civ Inf Eng 32(5):361–378

Chakraborty S (2021) Transfer learning based multi-fidelity physics informed deep neural network. J Comput Phys 426:109942

Chan S, Elsheikh AH (2018) A machine learning approach for efficient uncertainty quantification using multiscale methods. J Comput Phys 354:493–511

Chandrasekhar A, Suresh K (2021) TOuNN: Topology optimization using neural networks. Struct Multidisc Optim 63(3):1135–1149

Chen W, Ahmed F (2021a) MO-PaDGAN: Reparameterizing Engineering Designs for augmented multi-objective optimization. Appl Soft Comput 113:107909

Chen W, Ahmed F (2021b) Padgan: Learning to generate high-quality novel designs. J Mech Des 143(3):031703

Chen CT, Gu GX (2020) Generative deep neural networks for inverse materials design using backpropagation and active learning. Adv Sci 7(5):1902607

Chen X, Chen X, Zhou W, Zhang J, Yao W (2020) The heat source layout optimization using deep learning surrogate modeling. Struct Multidisc Optim 62(6):3127–3148

Chen W, Ahmed F (2020) PaDGAN: A generative adversarial network for performance augmented diverse designs. In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (Vol. 84003, p. V11AT11A010). American Society of Mechanical Engineers.

Chen T, Guestrin C (2016) Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (pp. 785–794)

Chen W, Chiu K, Fuge M (2019) Aerodynamic design optimization and shape exploration using generative adversarial networks. In AIAA Scitech 2019 Forum (p. 2351)

Chhabra JP, Warn GP (2019) A method for model selection using reinforcement learning when viewing design as a sequential decision process. Struct Multidisc Optim 59(5):1521–1542

Chi H, Zhang Y, Tang TLE, Mirabella L, Dalloro L, Song L, Paulino GH (2021) Universal machine learning for topology optimization. Comput Methods Appl Mech Eng 375:112739

Cortes C, Vapnik V (1995) Support-Vector Networks. Mach Learn 20(3):273–297

Cox DR (1958) The regression analysis of binary sequences. J Roy Stat Soc: Ser B (methodol) 20(2):215–232

Cutajar K, Osborne M, Cunningham J, Filippone M (2016) Preconditioning kernel matrices. In International conference on machine learning (pp. 2529–2538). PMLR.

Dai Y, Li Y, Liu LJ (2019) New product design with automatic scheme generation. Sens Imag 20(1):1–16

Deng C, Wang Y, Qin C, Lu W (2020) Self-directed online machine learning for topology optimization. arXiv preprint arXiv:2002.01927.

Deng H, To AC (2020) Topology optimization based on deep representation learning (DRL) for compliance and stress-constrained design. Comput Mech 66:449–469

Dering M, Cunningham J, Desai R, Yukish MA, Simpson TW, Tucker CS (2018) A physics-based virtual environment for enhancing the quality of deep generative designs. In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (Vol. 51753, p. V02AT03A015). American Society of Mechanical Engineers.

Diego-Mas JA, Alcaide-Marzal J (2016) Single users' affective responses models for product form design. Int J Ind Ergon 53:102–114

Doi S, Sasaki H, Igarashi H (2019) Multi-objective topology optimization of rotating machines using deep learning. IEEE Trans Magn 55(6):1–5

Dong K, Eriksson D, Nickisch H, Bindel D, Wilson AG (2017) Scalable log determinants for Gaussian process kernel learning. arXiv preprint arXiv:1711.03481

Du X, Xu H, Zhu F (2021) A data mining method for structure design with uncertainty in design variables. Comput Struct 244:106457

Džeroski S, Ženko B (2004) Is combining classifiers with stacking better than selecting the best one? Mach Learn 54(3):255–273

Elingaard MO, Aage N, Bærentzen JA, Sigmund O (2022) De-homogenization using convolutional neural networks. Comput Methods Appl Mech Eng 388:114197

Emmert-Streib F, Yang Z, Feng H, Tripathi S, Dehmer M (2020) An introductory review of deep learning for prediction models with big data. Front Artif Intel 3:4

Falck R, Gray JS, Ponnapalli K, Wright T (2021) dymos: A Python package for optimal control of multidisciplinary systems. J Open Source Soft 6(59):2809

Fernández-Godino MG, Park C, Kim NH, Haftka RT (2016) Review of multi-fidelity models. arXiv preprint arXiv:1609.07196.

Ferreiro-Cabello J, Fraile-Garcia E, de Pison Ascacibar EM, Martinez-de-Pison FJ (2018) Metamodel-based design optimization of structural one-way slabs based on deep learning neural networks to reduce environmental impact. Eng Struct 155:91–101

Fisher RA (1936) The use of multiple measurements in taxonomic problems. Ann Eugen 7(2):179–188

Fix E, Hodges JL (1989) Discriminatory analysis. Nonparametric discrimination: Consistency properties. Inter Stat Rev/revue Internationale De Statistique 57(3):238–247

Forrester A, Sobester A, Keane A (2008) Engineering design via surrogate modelling: a practical guide. John Wiley & Sons, USA

Freiesleben J, Keim J, Grutsch M (2020) Machine learning and design of experiments: Alternative approaches or complementary methodologies for quality improvement? Qual Reliab Eng Int 36(6):1837–1848

Friedman JH (2001) Greedy function approximation: a gradient boosting machine. Annal Stat 25:1189–1232

Fukushima K (1988) Neocognitron: A hierarchical neural network capable of visual pattern recognition. Neural Netw 1(2):119–130

García-Segura T, Yepes V, Frangopol DM (2017) Multi-objective design of post-tensioned concrete road bridges using artificial neural networks. Struct Multidisc Optim 56(1):139–150

Gardner JR, Pleiss G, Bindel D, Weinberger KQ, Wilson AG (2018a). Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. arXiv preprint arXiv:1809.11165.

Gardner J, Pleiss G, Wu R, Weinberger K, Wilson A (2018b) Product kernel interpolation for scalable Gaussian processes. In International Conference on Artificial Intelligence and Statistics (pp. 1407–1416). PMLR.

Garriga AG, Mainini L, Ponnusamy SS (2019) A machine learning enabled multi-fidelity platform for the integrated design of aircraft systems. J Mech Des 141(12):121405

Gladstone RJ, Nabian MA, Keshavarzzadeh V, Meidani H (2021) Robust topology optimization using variational autoencoders. arXiv preprint arXiv:2107.10661.

Goel T, Haftka RT, Shyy W, Queipo NV (2007) Ensemble of Surrogates. Struct Multidisc Optim 33(3):199–216

Golub GH, Reinsch C (1971) Singular value decomposition and least squares solutions. In Linear algebra (pp. 134–151). Springer, Berlin, Heidelberg

Gomes WJDS (2020) Shallow and deep artificial neural networks for structural reliability analysis. ASME J Risk Uncertainty Part B 6(4):041006

Gomes GSDS, Ludermir TB (2013) Optimization of the weights and asymmetric activation function family of neural network for time series forecasting. Expert Syst Appl 40(16):6438–6446

Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Bengio Y (2014) Generative adversarial nets. Adv Neur Info Process Syst 27.

Gray JS, Hwang JT, Martins JR, Moore KT, Naylor BA (2019) OpenMDAO: An open-source framework for multidisciplinary design, analysis, and optimization. Struct Multidisc Optim 59(4):1075–1104

Harman HH (1976) Modern factor analysis. University of Chicago press, USA

Harzing AW (2007). Publish or Perish, available from https://harzing.com/resources/publish-or-perish

Hasegawa K, Fukami K, Murata T, Fukagata K (2020) CNN-LSTM based reduced order modeling of two-dimensional unsteady flows around a circular cylinder at different Reynolds numbers. Fluid Dyn Res 52(6):065501

He L, Qian W, Zhao T, Wang Q (2020a) Multi-fidelity aerodynamic data fusion with a deep neural network modeling method. Entropy 22(9):1022

He P, Mader CA, Martins JR, Maki KJ (2020b) Dafoam: an open-source adjoint framework for multidisciplinary design optimization with openfoam. AIAA J 58(3):1304–1319

Hou TY, Lam KC, Zhang P, Zhang S (2019) Solving Bayesian inverse problems from the perspective of deep generative networks. Comput Mech 64(2):395–408

Jabarullah Khan NK, Elsheikh AH (2019) A machine learning based hybrid multi-fidelity multi-level Monte Carlo method for uncertainty quantification. Front Environ Sci 7:105

Janda T, Zemanová A, Hála P, Konrád P, Schmidt J (2020) Reduced order model of glass plate loaded by low-velocity impact. Int J Comput Methods Exp Meas 8(1):36–46

Jang S, Kang N (2020) Generative design by reinforcement learning: Maximizing diversity of topology optimized designs. arXiv preprint arXiv:2008.07119.

Jiang J, Fan JA (2019) Global optimization of dielectric metasurfaces using a physics-driven neural network. Nano Lett 19(8):5366–5372

Jiang J, Fan JA (2020) Simulator-based training of generative neural networks for the inverse design of metasurfaces. Nanophotonics 9(5):1059–1069

Jiang X, Wang H, Li Y, Mo K (2020) Machine learning based parameter tuning strategy for MMC based topology optimization. Adv Eng Softw 149:102841

Jin SS (2020) Compositional kernel learning using tree-based genetic programming for Gaussian process regression. Struct Multidisc Optim 62:1313–1351

Jung J, Yoon K, Lee PS (2020) Deep learned finite elements. Comput Methods Appl Mech Eng 372:113401

Kallioras NA, Lagaros ND (2020) DzAI𝕟: Deep learning based generative design. Procedia Manufacturing 44:591–598

Kallioras NA, Kazakis G, Lagaros ND (2020) Accelerated topology optimization by means of deep learning. Struct Multidisc Optim 62(3):1185–1212

Kambampati S, Du Z, Chung H, Kim HA, Jauregui C, Townsend S, Hedges L (2018). OpenLSTO: Open-source software for level set topology optimization. In: 2018 Multidisciplinary Analysis and Optimization Conference (p. 3882).

Kaplan EM, Acar E, Bülent Özer M (2021) Development of a method for maximum structural response prediction of a store externally carried by a jet fighter. Proce Inst Mech Eng Part G: J Aeros Eng 09544100211022244.

Karlik B, Olgac AV (2011) Performance analysis of various activation functions in generalized MLP architectures of neural networks. Int J Artif Intel Exp Sys 1(4):111–122

Karniadakis GE, Kevrekidis IG, Lu L, Perdikaris P, Wang S, Yang L (2021) Physics-informed machine learning. Nature Reviews. Physics 3(6):422–440

Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, Liu TY (2017) Lightgbm: A highly efficient gradient boosting decision tree. Adv Neural Inf Process Syst 30:3146–3154

Keshavarzzadeh V, Alirezaei M, Tasdizen T, Kirby RM (2021) Image-based multiresolution topology optimization using deep disjunctive normal shape model. Comput Aided Des 130:102947

Khan S, Gunpinar E, Moriguchi M, Suzuki H (2019a) Evolving a psycho-physical distance metric for generative design exploration of diverse shapes. J Mech Des 141(11):111101

Khan S, Gunpinar E, Sener B (2019b) GenYacht: An interactive generative design system for computer-aided yacht hull design. Ocean Eng 191:106462

Khatouri H, Benamara T, Breitkopf P, Demange J, Feliot P (2020) Constrained multi-fidelity surrogate framework using Bayesian optimization with non-intrusive reduced-order basis. Adv Model Simul Eng Sci 7(1):1–20

Kim SH, Boukouvala F (2020) Machine learning-based surrogate modeling for data-driven optimization: A comparison of subset selection for regression techniques. Optim Lett 14(4):989–1010

Kingma DP, Welling M (2013) Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.

Kochkov D, Smith JA, Alieva A, Wang Q, Brenner MP, Hoyer S (2021) Machine learning–accelerated computational fluid dynamics. Proc Nat Acad Sci 118(21):e2101784118

Kollmann HT, Abueidda DW, Koric S, Guleryuz E, Sobh NA (2020) Deep learning for topology optimization of 2D metamaterials. Mater Des 196:109098

Kou J, Zhang W (2019) A hybrid reduced-order framework for complex aeroelastic simulations. Aerosp Sci Technol 84:880–894

Kumar M, Yadav N (2011) Multilayer perceptrons and radial basis function neural network methods for the solution of differential equations: A survey. Comput Math Appl 62(10):3796–3811

Lafage R, Defoort S, Lefebvre T (2019) WhatsOpt: a web application for multidisciplinary design analysis and optimization. In AIAA Aviation 2019 Forum (p. 2990).

Lee J, Jeong H, Kang S (2008) Derivative and GA-based methods in metamodeling of back-propagation neural networks for constrained approximate optimization. Struct Multidisc Optim 35(1):29–40

Lee XY, Balu A, Stoecklein D, Ganapathysubramanian B, Sarkar S (2019) A case study of deep reinforcement learning for engineering design: Application to microfluidic devices for flow sculpting. J Mech Des 141(11):111401

Lee S, Kim H, Lieu QX, Lee J (2020) CNN-based image recognition for topology optimization. Knowl-Based Syst 198:105887

Lee YO, Jo J, Hwang J (2017). Application of deep neural network and generative adversarial network to industrial maintenance: A case study of induction motor fault detection. In: 2017 IEEE international conference on big data (big data) (pp. 3248–3253). IEEE

Lee M, Park Y, Jo H, Kim K, Lee S, Lee I (2022) Deep generative tread pattern design framework for efficient conceptual design. J Mechan Des 1–28.

Lei X, Liu C, Du Z, Zhang W, Guo X (2019) Machine learning-driven real-time topology optimization under moving morphable component-based framework. J Appl Mech 86(1):011004

Li Y, Mei F (2021) Deep learning-based method coupled with small sample learning for solving partial differential equations. Mult Tools Appl 80(11):17391–17413

Li M, Wang Z (2021) An LSTM-based ensemble learning approach for time-dependent reliability analysis. J Mech Des 143(3):031702

Li B, Huang C, Li X, Zheng S, Hong J (2019) Non-iterative structural topology optimization using deep learning. Comput Aided Des 115:172–180

Li S, Xing W, Kirby R, Zhe S (2020) Multi-fidelity Bayesian optimization via deep neural networks. Adv Neural Info Proc Syst 33.

Liao H, Zhang W, Dong X, Poczos B, Shimada K, Burak Kara L (2020) A deep reinforcement learning approach for global routing. J Mech Des 142(6):061701

Lin Q, Hong J, Liu Z, Li B, Wang J (2018) Investigation into the topology optimization for conductive heat transfer based on deep learning approach. Int Commun Heat Mass Transfer 97:103–109

Lin Q, Liu Z, Hong J (2019) Method for directly and instantaneously predicting conductive heat transfer topologies by using supervised deep learning. Int Commun Heat Mass Transfer 109:104368

Liu K, Tovar A, Nutwell E, Detwiler D (2015) Thin-walled compliant mechanism component design assisted by machine learning and multiple surrogates.

Liu D, Wang Y (2019) Multi-fidelity physics-constrained neural network and its application in materials modeling. J Mech Des 141(12):121403

Lye KO, Mishra S, Ray D, Chandrashekar P (2021) Iterative surrogate model optimization (ISMO): An active learning algorithm for PDE constrained optimization with deep neural networks. Comput Methods Appl Mech Eng 374:113575

Lynch ME, Sarkar S, Maute K (2019) Machine learning to aid tuning of numerical parameters in topology optimization. J Mech Des 141(11):114502

Ma SB, Kim S, Kim JH (2020) Optimization design of a two-vane pump for wastewater treatment using machine-learning-based surrogate modeling. Processes 8(9):1170

McCulloch WS, Pitts W (1943) A logical calculus of the ideas immanent in nervous activity. Bull Math Biophys 5(4):115–133

McFall KS (2013) Automated design parameter selection for neural networks solving coupled partial differential equations with discontinuities. J Franklin Inst 350(2):300–317

Meng X, Karniadakis GE (2020) A composite neural network that learns from multi-fidelity data: Application to function approximation and inverse PDE problems. J Comput Phys 401:109020

Minisci E, Vasile M (2013) Robust design of a reentry unmanned space vehicle by multifidelity evolution control. AIAA J 51(6):1284–1295

Mondal S (2020) Probabilistic machine learning for advanced engineering design optimization and diagnostics, PhD dissertation, Penn State University.

Montgomery DC, Peck EA, Vining GG (2021) Introduction to linear regression analysis. John Wiley & Sons, USA

Motamed M (2020) A multi-fidelity neural network surrogate sampling method for uncertainty quantification. Int J Uncertain Quantif 10(4).

Mozaffar M, Bostanabad R, Chen W, Ehmann K, Cao J, Bessa MA (2019) Deep learning predicts path-dependent plasticity. Proc Natl Acad Sci 116(52):26414–26420

Müller J, Park J, Sahu R, Varadharajan C, Arora B, Faybishenko B, Agarwal D (2021) Surrogate optimization of deep neural networks for groundwater predictions. J Global Optim 81(1):203–231

Nagarajan HP, Mokhtarian H, Jafarian H, Dimassi S, Bakrani-Balani S, Hamedi A, Haapala KR (2019) Knowledge-based design of artificial neural network topology for additive manufacturing process modeling: A new approach and case study for fused deposition modeling. J Mech Des 141(2):021705

Nakamura K, Suzuki Y (2020) Deep learning-based topological optimization for representing a user-specified design area. arXiv preprint arXiv:2004.05461.

Napier N, Sriraman SA, Tran HT, James KA (2020) An artificial neural network approach for generating high-resolution designs from low-resolution input in topology optimization. J Mech Des 142(1):011402

Naranjo-Pérez J, Infantes M, Jiménez-Alonso JF, Sáez A (2020) A collaborative machine learning-optimization algorithm to improve the finite element model updating of civil engineering structures. Eng Struct 225:111327

Nie Z, Lin T, Jiang H, Kara LB (2021) Topologygan: Topology optimization using generative adversarial networks based on physical fields over the initial domain. J Mech Des 143(3):031715

Ning C, You F (2018) Data-driven stochastic robust optimization: General computational framework and algorithm leveraging

machine learning for optimization under uncertainty in the big data era. Comput Chem Eng 111:115–133
Nobari AH, Rashad MF, Ahmed F (2021) Creativegan: Editing generative adversarial networks for creative design synthesis. arXiv preprint arXiv:2103.06242.
Odonkor P, Lewis K (2019) Data-driven design of control strategies for distributed energy systems. J Mech Des 141(11):111404
Oh S, Jung Y, Lee I, Kang N (2018) Design automation by integrating generative adversarial networks and topology optimization. In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (Vol. 51753, p. V02AT03A008). American Society of Mechanical Engineers.
Oh S, Jung Y, Kim S, Lee I, Kang N (2019) Deep generative design: Integration of topology optimization and generative models. J Mech Des 141(11).
Owoyele O, Pal P, Vidal Torreira A, Probst D, Shaxted M, Wilde M, Senecal PK (2021) An automated machine learning-genetic algorithm (AutoML-GA) approach for efficient simulation-driven engine design optimization. arXiv e-prints, arXiv-2101
Panchal JH, Fuge M, Liu Y, Missoum S, Tucker C (2019) Machine learning for engineering design. J Mech Des 141(11)
Pánek D, Orosz T, Karban P (2020) Artap: Robust design optimization framework for engineering applications. arXiv 2019. arXiv preprint arXiv:1912.11550
Parsonage B, Maddock CA (2020) Multi-stage multi-fidelity information correction for artificial neural network based meta-modelling. In 2020 IEEE Symposium Series on Computational Intelligence (SSCI) (pp. 950–957). IEEE
Patel J, Choi SK (2012) Classification approach for reliability-based topology optimization using probabilistic neural networks. Struct Multidisc Optim 45(4):529–543
Pawar S, Rahman SM, Vaddireddy H, San O, Rasheed A, Vedula P (2019) A deep learning enabler for nonintrusive reduced order modeling of fluid flows. Phys Fluids 31(8):085101
Peherstorfer B, Willcox K, Gunzburger M (2018) Survey of multifidelity methods in uncertainty propagation, inference, and optimization. SIAM Rev 60(3):550–591
Pereira DR, Piteri MA, Souza AN, Papa JP, Adeli H (2020) FEMa: A finite element machine for fast learning. Neural Comput Appl 32(10):6393–6404
Perez RE, Jansen PW, Martins JR (2012) pyOpt: a Python-based object-oriented framework for nonlinear constrained optimization. Struct Multidisc Optim 45(1):101–118
Perron C, Rajaram D, Mavris DN (2021) Multi-fidelity non-intrusive reduced-order modelling based on manifold alignment. Proce Royal Soc A 477(2253):20210495
Pillai AC, Thies PR, Johanning L (2019) Mooring system design optimization using a surrogate assisted multi-objective genetic algorithm. Eng Optim 51(8):1370–1392
Popov AA, Mou C, Sandu A, Iliescu T (2021) A multifidelity ensemble Kalman filter with reduced order control variates. SIAM J Sci Comput 43(2):A1134–A1162
Puentes L, Raina A, Cagan J, McComb C. (2020) Modeling a strategic human engineering design process: Human-inspired heuristic guidance through learned visual design agents. In Proceedings of the Design Society: DESIGN Conference (Vol. 1, pp. 355–364). Cambridge University Press.
Qian C, Ye W (2021) Accelerating gradient-based topology optimization design with dual-model artificial neural networks. Struct Multidisc Optim 63(4):1687–1707
Radford A, Metz L, Chintala S (2015) Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434.
Raina A, McComb C, Cagan J (2019). Learning to design from humans: Imitating human designers through deep learning. J Mech Des 141(11)
Raissi M, Karniadakis GE (2018) Hidden physics models: Machine learning of nonlinear partial differential equations. J Comput Phys 357:125–141
Raissi M, Perdikaris P, Karniadakis GE (2019) Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J Comput Phys 378:686–707
Rasmussen CE (2003). Gaussian processes in machine learning. In Summer school on machine learning (pp. 63–71). Springer, Berlin, Heidelberg
Rätsch G, Onoda T, Müller KR (2001) Soft Margins for AdaBoost. Mach Learn 42(3):287–320
Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. Nature 323(6088):533–536
Sasaki H, Igarashi H (2019a) Topology optimization accelerated by deep learning. IEEE Trans Magn 55(6):1–5
Sasaki H, Igarashi H (2019b) Topology optimization of IPM motor with aid of deep learning. Int J Appl Electromagnet Mech 59(1):87–96
Shi M, Lv L, Sun W, Song X (2020a) A multi-fidelity surrogate model based on support vector regression. Struct Multidisc Optim 61(6):2363–2375
Shi X, Qiu T, Wang J, Zhao X, Qu S (2020b) Metasurface inverse design using machine learning approaches. J Phys D Appl Phys 53(27):275105
Shu D, Cunningham J, Stump G, Miller SW, Yukish MA, Simpson TW, Tucker CS (2020) 3d design using generative adversarial networks and physics-based validation. J Mech Des 142(7):071701
Singh AP, Medida S, Duraisamy K (2017) Machine-learning-augmented predictive modeling of turbulent separated flows over airfoils. AIAA J 55(7):2215–2227
Singla M, Ghosh D, Shukla KK (2020) A survey of robust optimization based machine learning with special reference to support vector machines. Int J Mach Learn Cybern 11(7):1359–1385
Solanki KN, Acar E, Rais-Rohani M, Horstemeyer MF, Steele WG (2009) Product design optimisation with microstructure-property modelling and associated uncertainties. Int J Des Eng 2(1):47–79
Song H, Choi KK, Lee I, Zhao L, Lamb D (2013) Adaptive virtual support vector machine for reliability analysis of high-dimensional problems. Struct Multidisc Optim 47(4):479–491
Sosnovik I, Oseledets I (2019) Neural networks for topology optimization. Russ J Numer Anal Math Model 34(4):215–223
Strömberg N (2020) Efficient detailed design optimization of topology optimization concepts by using support vector machines and metamodels. Eng Optim 52(7):1136–1148
Su G, Peng L, Hu L (2017) A Gaussian process-based dynamic surrogate model for complex engineering structural reliability analysis. Struct Saf 68:97–109
Sun H, Ma L (2020) Generative design by using exploration approaches of reinforcement learning in density-based structural topology optimization. Designs 4(2):10
Sun G, Wang S (2019) A review of the artificial neural network surrogate modeling in aerodynamic design. Proc Inst Mech Eng, Part G: J Aeros Eng 233(16):5863–5872
Sutton RS, Barto AG (2018) Reinforcement learning: An introduction. MIT press.
Tan RK, Zhang NL, Ye W (2020) A deep learning-based method for the design of microstructural materials. Struct Multidisc Optim 61(4):1417–1438
Tao J, Sun G (2019) Application of deep learning based multi-fidelity surrogate model to robust aerodynamic design optimization. Aerosp Sci Technol 92:722–737

Tenne Y (2019). Enhancing simulation-driven optimization by machine-learning. Int J Model Optim 9(4)
Thole SP, Ramu P (2020) Design space exploration and optimization using self-organizing maps. Struct Multidisc Optim 62(3):1071–1088
Trehan S, Carlberg KT, Durlofsky LJ (2017) Error modeling for surrogates of dynamical systems using machine learning. Int J Numer Meth Eng 112(12):1801–1827
Trinchero R, Larbi M, Torun HM, Canavero FG, Swaminathan M (2018) Machine learning and uncertainty quantification for surrogate models of integrated devices with a large number of parameters. IEEE Access 7:4056–4066
Tripathy RK, Bilionis I (2018) Deep UQ: Learning deep neural network surrogate models for high dimensional uncertainty quantification. J Comput Phys 375:565–588
Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Polosukhin I (2017) Attention is all you need. In Advances in neural information processing systems (pp. 5998–6008)
Wang C, Yao S, Wang Z, Hu J (2021a) Deep super-resolution neural network for structural topology optimization. Eng Optim 53(12):2108–2121
Wang D, Xie C, Wang S (2021c) An adaptive RBF neural network–based multi-objective optimization method for lightweight and crashworthiness design of cab floor rails using fuzzy subtractive clustering algorithm. Struct Multidisc Optim 63(2):915–928
Wang L, van Beek A, Da D, Chan YC, Zhu P, Chen W (2022) Data-driven multiscale design of cellular composites with multiclass microstructures for natural frequency maximization. Compos Struct 280:114949
Wang F, Song M, Edelen A, Huang X (2019) Machine learning for design optimization of storage ring nonlinear dynamics. arXiv preprint arXiv:1910.14220.
Wang D, Xiang C, Pan Y, Chen A, Zhou X, Zhang Y (2021b) A deep convolutional neural network for topology optimization with perceptible generalization ability. Eng Optim 1–16
Wiener N (1938) The homogeneous chaos. Am J Math 60(4):897–936
Williams CK, Rasmussen CE (2006) Gaussian processes for machine learning (Vol. 2, No. 3, p. 4). Cambridge, MA: MIT press.
Williams G, Meisel NASimpson TW, McComb C (2019) Design repository effectiveness for 3D convolutional neural networks: Application to additive manufacturing. J Mechan Des 141(11)
Wu X, Kozlowski T, Meidani H (2018) Kriging-based inverse uncertainty quantification of nuclear fuel performance code BISON fission gas release model using time series measurement data. Reliab Eng Syst Saf 169:422–436
Wu J (2017) Introduction to convolutional neural networks. National Key Lab for Novel Software Technology. Nanjing University. China 5(23), 495.
Wuraola A, Patel N (2018) SQNL: A new computationally efficient activation function. In: 2018 International Joint Conference on Neural Networks (IJCNN) (pp. 1–7). IEEE
Xu Y, Gao Y, Wu C, Fang J, Sun G, Steven GP, Li Q (2021) Machine learning based topology optimization of fiber orientation for variable stiffness composite structures. Int J Num Methods Eng
Yamasaki S, Yaji K, Fujita K (2021) Data-driven topology design using a deep generative model. Struct Multidisc Optim 1–20.
Yan L, Zhou T (2019). An adaptive surrogate modeling based on deep neural networks for large-scale Bayesian inverse problems. arXiv preprint arXiv:1911.08926.
Yang Y, Perdikaris P (2019) Conditional deep surrogate models for stochastic, high-dimensional, and multi-fidelity systems. Comput Mech 64(2):417–434
Yao H, Gao Y, Liu Y (2020) FEA-Net: A physics-guided data-driven model for efficient mechanical response prediction. Comput Methods Appl Mech Eng 363:112892
Yonekura K, Suzuki K (2021) Data-driven design exploration method using conditional variational autoencoder for airfoil design. Struct Multidisc Optim 1–12.
Yonekura K, Hattori H (2019) Framework for design optimization using deep reinforcement learning. Struct Multidisc Optim 60(4):1709–1713
Yu Y, Hur T, Jung J, Jang IG (2019) Deep learning for determining a near-optimal topological design without any iteration. Struct Multidisc Optim 59(3):787–799
Yuan C, Moghaddam M (2020) Attribute-aware generative design with generative adversarial networks. IEEE Access 8:190710–190721
Zhang Y, Ye W (2019) Deep learning-based inverse method for layout design. Struct Multidisc Optim 60(2):527–536
Zhang J, Zhao X (2021) Machine-learning-based surrogate modeling of aerodynamic flow around distributed structures. AIAA J 59(3):868–879
Zhang X, Xie F, Ji T, Zhu Z, Zheng Y (2021a) Multi-fidelity deep neural network surrogate model for aerodynamic shape optimization. Comput Methods Appl Mech Eng 373:113485
Zhang Z, Li Y, Zhou W, Chen X, Yao W, Zhao Y (2021b) TONR: An exploration for a novel way combining neural network with topology optimization. Comput Method Appl Mech Eng 386:114083