Neurocomputing 483 (2022) 42–58

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Survey paper

Evolutionary deep learning: A survey

Zhi-Hui Zhan [a,*], Jian-Yu Li [a], Jun Zhang [b]

[a] School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China
[b] Hanyang University, Ansan 15588, South Korea

ARTICLE INFO

Article history:
Received 2 November 2021
Revised 28 December 2021
Accepted 14 January 2022
Available online 3 February 2022

Keywords:
Deep learning
Evolutionary computation
Evolutionary algorithm
Swarm intelligence
Evolutionary deep learning
Artificial intelligence

ABSTRACT

As an advanced artificial intelligence technique for solving learning problems, deep learning (DL) has achieved great success in many real-world applications and attracted increasing attention in recent years. However, as the performance of DL depends on many factors such as the architecture and hyperparameters, how to optimize DL has become a hot research topic in the field of DL and artificial intelligence. Evolutionary computation (EC), including evolutionary algorithm and swarm intelligence, is a kind of efficient and intelligent optimization methodology inspired by the mechanisms of biological evolution and behaviors of swarm organisms. Therefore, a large number of researches have proposed EC algorithms to optimize DL, so called evolutionary deep learning (EDL), which have obtained promising results. Given the great progress and rapid development of EDL in recent years, it is quite necessary to review these developments in order to summarize previous research experiences and knowledge, as well as provide references to benefit the development of more researches and applications. For this aim, this paper categorizes existing works in a two-level taxonomy. The higher level includes four categories based on *when* the EC can be adopted in optimizing the DL, which are the four procedures of the whole DL lifetime, including *data processing, model search, model training,* and *model evaluation and utilization*. In the lower level, related works in each category are further classified according to the functionality and the aim of using EC in the corresponding DL procedure, i.e., *why* using EC in this DL procedure. As a result, the taxonomy can clearly show *how* an EC algorithm can be used to optimize and improve DL. Moreover, this survey also discusses the potential research directions to provide the prospect of EDL in the future.

1. Introduction

As one of the representative artificial intelligence (AI) and machine learning (ML) technologies, deep learning (DL) has obtained rapid development and attracted increasing attention in recent years [1,2]. Different from other ML technologies, DL usually uses the deep neural network (DNN) model to tackle learning problems including classification and prediction [2,3]. Compared to shallow models, deeper models can have a much stronger learning ability to learn more useful features to meet the need of various complex learning tasks. Moreover, by using their large and deep structure, DL models can extract knowledge from big data for real-world applications. As a result, many researches and studies acrossing various fields have paid increasing attention to DL and DL-related methods and applications [4–7].

However, although various DL methods have been proposed and successfully applied in many applications, the performance of DL can be influenced by many factors such as the architecture and hyperparameters, which involves data quality, model configuration and robustness, learning techniques, and problem characteristics. Therefore, it is not easy to improve DL performance on accuracy and efficiency, especially for real-world problems that have insufficient prior knowledge and information [8–11]. For example, designing DL models with better architectures and hyperparameters for image classification tasks usually requires not only the rich knowledge in DL and image processing fields but also the numerous and tedious trial-and-error experiments, which is difficult for users from other fields to modify and extend for their needs [12]. Therefore, many researches have studied more intelligent methods to improve and optimize DL automatically according to the problems, where evolutionary computation (EC) methods are usually adopted [12–14].

EC is also an important branch in AI that is efficient in optimizing complex optimization problems [15,16]. In general, EC imitates the evolution rule of ‘‘survival of the fittest” from nature to evolve candidate solutions, so as to obtain more satisfactory solutions for optimization problems. Generally speaking, EC algorithms mainly

* Corresponding author.
E-mail address: zhanapollo@163.com (Z.-H. Zhan).

https://doi.org/10.1016/j.neucom.2022.01.099
0925-2312/

include evolutionary algorithm (EA) [17] and swarm intelligence (SI) [18] algorithms. To be more specific, representative EAs include genetic algorithm (GA) [19,20], evolutionary strategy (ES) [21], genetic programming (GP) [22], evolutionary programming (EP) [23], differential evolution (DE) [24–27], and estimation of distribution algorithm (EDA) [28,29], while representative SI algorithms include ant colony optimization (ACO) [30–32] and particle swarm optimization (PSO) [33–37]. Moreover, the combination of EA and SI algorithms have also been studied and proposed, such as the whale optimization algorithm and DE [38].

In the AI community, as ML (e.g., DL) and EC are respectively the efficient learning and optimization technologies, their developments are tightly integrated and greatly benefited from each other [39–42]. To date, the research direction of both using ML to enhance EC and using EC to optimize ML have yielded fruitful encouraging results [40–44]. Moreover, as the DL developed dramatically in recent years, the researches into improving DL via EC have increased rapidly and made great progress [45–48]. The branch of using EC to improve DL is called evolutionary deep learning (EDL). To better show the dramatical increase of EDL works, Fig. 1 plots the published work about EDL in different databases from 2010 to 2020, where the search strategies in the corresponding search engines are as:

i. IEEE Xplore: "All Metadata": evolutionary deep learning;
ii. Web of science: evolutionary deep learning (Topic);
iii. Scopus: TITLE-ABS-KEY (evolutionary AND deep AND learning); and
iv. Google scholar: intitle: evolutionary deep learning.

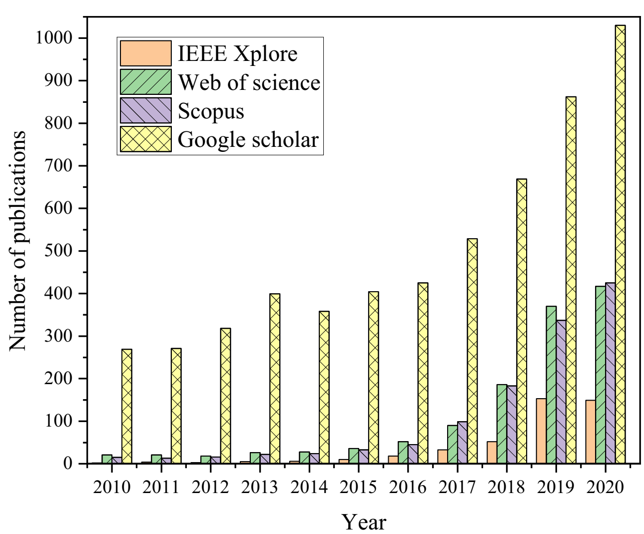

Fig. 1. The number of publications about EDL from 2010 to 2020 in different databases.

As can be seen in Fig. 1, the number of publications about EDL have increased rapidly and obviously since 2015, which is close to the occurrence time of some representative DNNs in 2015 [49] and the news of famous DL applications like that the AlphaGo beat the world champion Lee Sedol in 2016 [50].

Given the rapid and significant advances in EDL, it is quite necessary to review these developments at this time in order to summarize previous research experiences, as well as provide references to promote the development of more researches and applications. Although there have been some reviews about EDL [47,48], these surveys focus on EC methods for some specific issues in DL, like EDL about model architecture search and hyperparameter optimization, while some other EDL works about other procedures in the DL lifetime are still scattered in the literature and need to be consolidated systematically.

Therefore, this survey attempts to provide a systematic review of existing EDL work in the whole lifetime of using DL for solving problems, which includes EC for not only the architecture and hyperparameter optimization but also other procedures for improving DL, so as to completely review and analyze how EC can improve DL efficiently. To better illustrate the differences between existing surveys and this survey, Table 1 compares different surveys in terms of "*perspective and focus*" and "*taxonomy and category*". As can be seen, this survey has a larger perspective and more systematic taxonomy than existing surveys. To be specific, the taxonomy in this paper can be divided into two levels. The higher level is according to the four major procedures in the whole lifetime when using DL for solving problems, namely, *data processing, model search, model training,* and *model evaluation and utilization*, which can show that when and how EC can improve DL through different procedures of the whole lifetime. Following this, the lower level is based on the functionality and aim of using EC in each procedure of EDL. Consequently, the taxonomy can provide a clear picture to show how an EC algorithm can be used to optimize and improve DL. Besides, some potential future research directions are also discussed in this survey, so as to inspire more researches and applications.

The following contents are organized as follows. Section 2 provides preliminaries including the introduction of DL models and the taxonomy of this survey. Section 3–6 reviews existing EDL works in order according to the taxonomy, while Section 7 discusses the potential future research directions. Finally, Section 8 draws the conclusion.

Table 1
Comparisons of Different Surveys about EC for DL.

Survey	Perspective and focus	Taxonomy and category	Year
Liu et al.' survey [44]	EC for neural architecture search	Encoding space/encoding strategy/population updating/efficient evaluation/application	2021
Zhou et al.' survey [45]	EC for the construction of DNNs	Optimization on architecture/parameter/both architecture and parameter/miscellaneous issues like different tasks and objectives	2021
Zhou et al.' survey [46]	advances of evolutionary neural architecture search	shallow and deep networks/and innovation/booming/trends of evolutionary DNNs	2021
Darwish et al.' survey [47]	EC for both architecture search and hyperparameter optimization	Evolutionary algorithm/swarm intelligence for DNN/CNN/DBN/RNN	2020
Bharti et al.' survey [48]	EC for both architecture search and hyperparameter optimization	EC for LSTM/CNN/GAN/AE/ESN	2020
This survey	EC in the whole DL procedure	EC for data process/model search/model training/model evaluation and utilization in the DL procedure	–

2. Preliminaries

2.1. Deep learning model

In general, widely-used DL models can be roughly divided into five categories according to their basic model, i.e., convolutional neural network (CNN) [8], deep belief network (DBN) [9–11], stacked auto-encoder (SAE) [12], recurrent neural network (RNN) [13], and generative adversarial network (GAN) [51,52]. These models are introduced briefly in the following. Note that as this part aims at introducing deep models while artificial neural network (ANN) often refers to shallow models in the literature [53,54], the ANN is not listed herein to avoid confusion and misunderstanding.

CNN is a widely-used DNN structure consisting of convolutional layers, pooling layers, and fully connected layers. Fig. 2(a) shows a common example of CNNs. As manually-designed CNNs have achieved great success in many well-known deep learning applications, e.g., image classification tasks, many researches have been conducted to obtain better CNNs, including the EC-based optimization methods, which will be reviewed in the following sections.

DBN can be developed by stacking restricted Boltzmann machines (RBMs), as shown in Fig. 2(b). In general, DBNs can pre-train the network in an unsupervised fashion and then fine-tunes it in a supervised way according to the labeled data, so as to obtain a more satisfactory features representation.

SAE is composed of multiple auto-encoders (AEs) in a stacking manner. Generally speaking, an AE consists of two symmetrical components (i.e., encoder and decoder), as shown in Fig. 2(c), and regards the raw input data as the idea output, so that the encoder and decoder can learn meaningful features to represent the raw data in the case that the learned features can be also transformed back to the raw data with minimum information loss.

RNN is a special neural network that contains recurrent connections. A time-expended structure example of RNNs is presented as Fig. 2(d), where the value of a hidden neuron at time t (i.e., h^t) will be influenced by both the value at time t-1 (i.e., h^{t-1}) and the output of its input. Due to the recurrent design, the weights parameters (e.g., U, W, and V in Fig. 2(d)) are shared in RNNs.

GAN is a deep learning model that contains two components, i.e., generator and discriminator, as illustrated in Fig. 2(e). The generator learns how to generate data similar to real training data to fool the discriminator, while the discriminator learns how to figure out the data from a real training set or generated by the generator. After training, the generator will have a great probability for data generation, which is meaningful in many real-world applications.

2.2. Taxonomy

In general, DL aims to learn meaningful and needed knowledge based on given data for handling corresponding tasks. Mathematically, the goal of DL is to achieve:

$$\max E_{test}(M, \underset{\theta_M}{\operatorname{argmax}} E_{train}(M, \theta_M, D_{train}), D_{test}) \tag{1}$$

where M is a DL model, θ_M represents the model parameters (e.g., weights) of M, D_{train} and D_{test} are the datasets for model training and test, respectively, and E_{train} and E_{test} are the evaluation criteria for model training and test, respectively. Note that the test data should not be used for training, so as to guarantee the evaluation of the generalization ability of the model. Therefore, the following inequation can be true even though the E_{test} is set as the same as E_{train}:

$$\begin{aligned} &\max E_{test}(M, \underset{\theta_M}{\operatorname{argmax}} E_{train}(M, \theta_M, D_{train}), D_{test}) \\ &= \max E_{test}(M, \underset{\theta_M}{\operatorname{argmax}} E_{test}(M, \theta_M, D_{train}), D_{test}) \\ &\neq \max E_{test}(M, \underset{\theta_M}{\operatorname{argmax}} E_{train}(M, \theta_M, D_{test}), D_{test}) \\ &= \max E_{test}(M, \underset{\theta_M}{\operatorname{argmax}} E_{test}(M, \theta_M, D_{test}), D_{test}) \end{aligned} \tag{2}$$

As can be seen, maximizing the performance of DL should consider four factors: data (i.e., D_{train} and D_{test}), model selection (i.e., M), model parameters (i.e., θ_M), and evaluation mechanism (i.e., E_{train} and E_{test}). Therefore, DL generally has four procedures for solving learning problems, which are data processing, model search, model training, and model evaluation and utilization, and thus better DL can be achieved by enhancing the results of one or some of these four procedures. Moreover, the key issues for enhancing the results of the four procedures can be summarized as four "how to", as shown in Fig. 3, i.e., how to obtain better data in the data processing procedure, how to find better models in the model search procedure, how to better train models to obtain suitable model parameters in the model training procedure, and how to better evaluate and utilize the model in the model evaluation and utilization procedure.

Based on the above, this survey categorizes and reviews the existing EDL works into four categories (i.e., four parts), so as to show how EC improves DL for solving learning problems from the perspective of the DL procedures. To improve the clarity and intuitiveness of the review, the corresponding taxonomy of this survey is illustrated in Fig. 4, which contains four parts. To be more specific, *the first part (*i.e., *data processing)* includes data generation, imbalanced data processing, and task-specific processing; *the second part (*i.e., *model search)* includes model architecture optimization, model hyperparameter optimization, and multi-objective model search; *the third part (*i.e., *model training)* includes model parameter optimization, model pre-training optimization, and training cost reduction; and *the fourth part (*i.e., *model evaluation and utilization)* includes model robustness evaluation, model ensemble, and model pruning. For the sake of concise and clarity, this paper picks and cites related significant works based on paper quality, impact, topic relevance, publication source, and publication time.

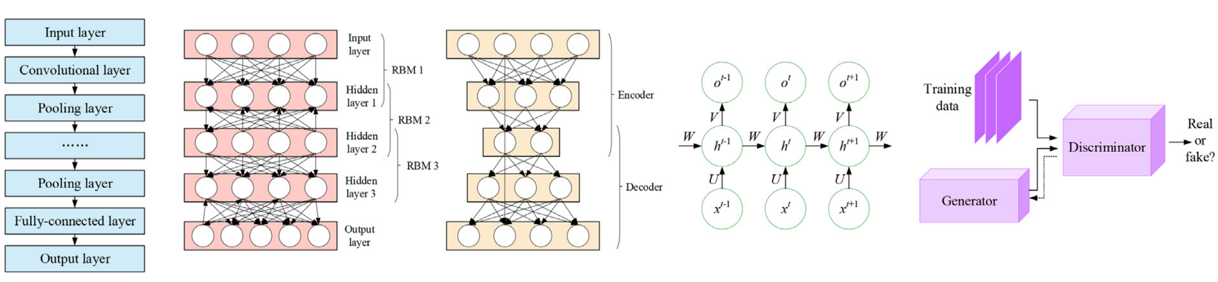

Fig. 2. Examples of some different deep learning models: (a) CNN; (b) DBN; (c) AE; (d) RNN; and (e) GAN.

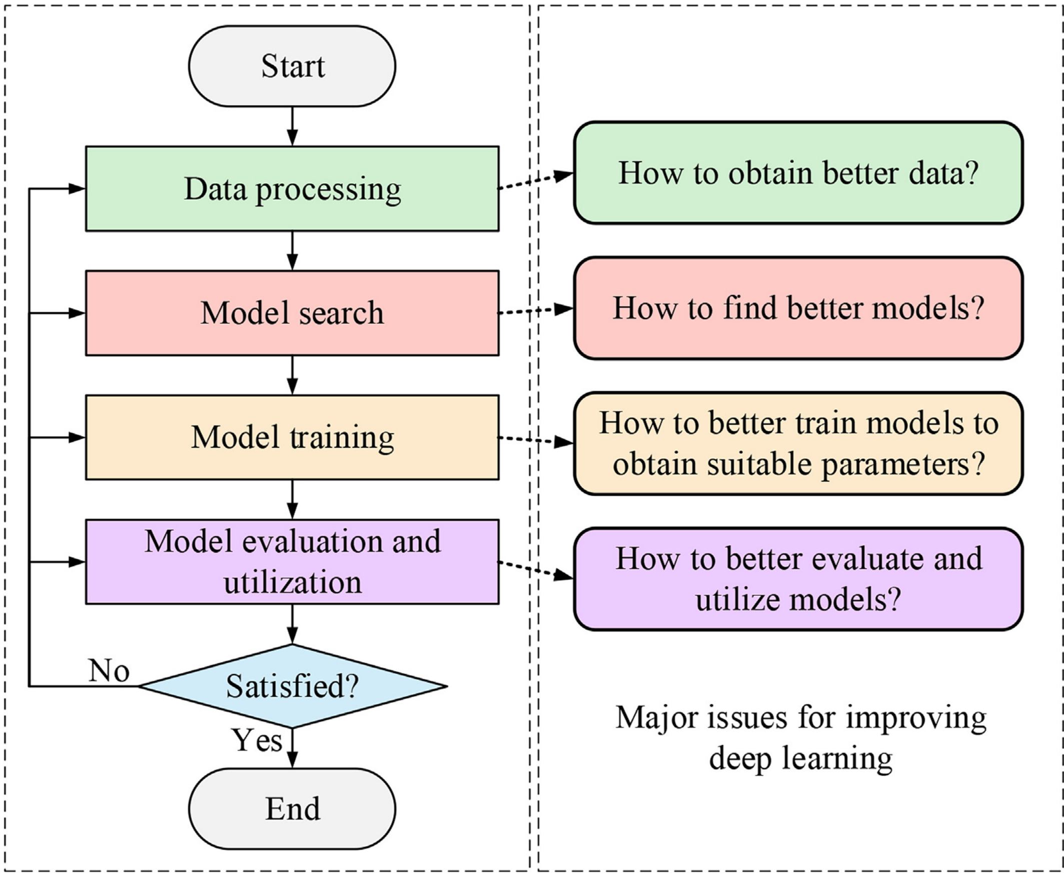

Fig. 3. The flowchart of the whole DL lifetime with major issues for improving DL in different procedures.

3. EC for data processing

As DL aims to learn knowledge from the data, the data quality will greatly influence the final results of DL. However, high-quality data are usually not sufficient in real-world applications, while available data may have various problems, e.g., incomplete, imbalance, noisy, and lack of labels, which require suitable data processing. Therefore, EDL can improve DL by optimizing the data processing via EC methods. Existing EDL works that focus on data processing can be roughly categorized into three categories, i.e., data generation, imbalance data processing, and task-specific data pre-processing, where the three categories and their corresponding issues are shown in Fig. 5. These three categories are introduced in the following contents.

3.1. Data generation

In practical applications, it is difficult to obtain high-quality training data samples with correct labels, which makes it very difficult to train accurate and efficient DL models. Therefore, many researches have studied using EC methods to generate high-quality training data and synthetic labels. To generate suitable data, GANs are usually helpful. Therefore, EC methods can be utilized to evolve better GANs, so as to help generate more accurate data. For example, Wang et al. [51] proposed an evolutionary GAN (E-GAN) framework for data generation. The proposed framework uses various adversarial training objectives as mutation operations to evolve diverse individual networks, which can achieve convincing generative performance on some large-scale datasets. Mehta et al. [55] designed an improved EA to evolve the best structure of GAN for data generation. Experimental results verified that compared with existing methods, this method can help generate similar but completely new data images which can be further used for training diagnostic neural networks. Considering the difficulty of insufficient labels, Dutta et al. [56] proposed a clustering technique based on multi-objective EAs to add weak labels to unlabeled data. Then the data with weak labels are fed to a GAN for training. Experimental studies show that this method can achieve good results on the real gene expression data set.

3.2. Imbalanced data processing

In practical application, data imbalance (e.g., data amount in different classes differs greatly) is a common problem, which will mislead the model to be prone to classes with more data and therefore affect the generalization ability of the model. To solve the imbalanced problem, approaches can be considered from two levels: the data-level approach and the algorithm-level approach.

The data-level approach mainly aims to reduce the imbalance degree in data amount based on data resampling methods, including increasing the samples of the minority class (i.e., over-sampling) and decreasing the samples of the majority class (i.e., under-sampling). As the over-sampling can be regarded as a data generation process for minority classes, the EC-based data generation optimization mentioned in the previous contents can be applied [51–56]. For example, Ma et al. [57] utilized EC to evolve the parameters in the safe-level synthetic minority over-sampling technique to generate data more properly. Different from over-sampling, Le et al. [58] considered the under-sampling as a binary optimization problem and proposed a surrogate-assisted evolutionary under-sampling method to solve the problem. Experimental results on 44 imbalanced datasets show that this evolu-

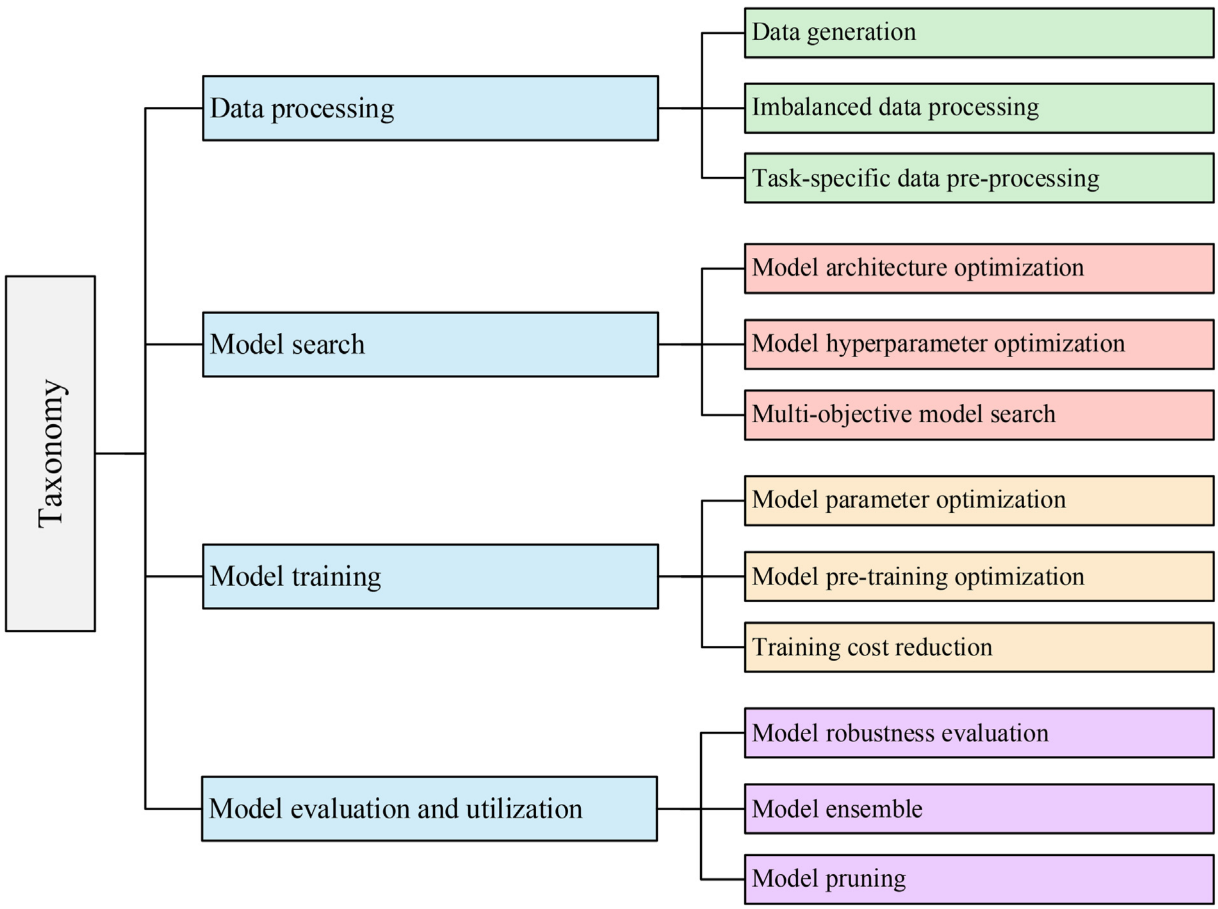

Fig. 4. An illustration of the taxonomy.

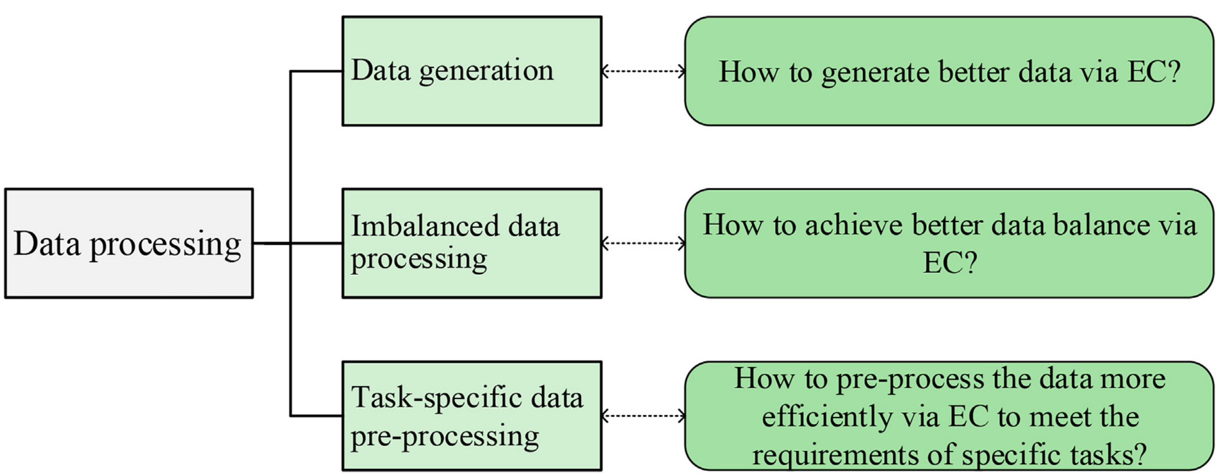

Fig. 5. Categories and corresponding issues of EDL in the data processing procedure.

tionary under-sampling method can help learning models obtain competitive classification performance and save up about 80% computational cost. Also for under-sampling, Roshan et al. [59] considered multiple objectives, such as sampling diversity, classification accuracy, and the number of sampled datasets, so as to propose an under-sampling method based on evolutionary multi-objective optimization. Experimental comparisons with state-of-the-art methods on over 33 imbalanced datasets have shown the superiority of the proposed under-sampling method. In addition, Zhu et al. [60] detected the overlapping regions between majority and minority classes and developed an EA to reserve the appropriate samples and to drop the rest unnecessary samples in the regions. Triguero et al. [61] attempted to develop an evolutionary under-sampling method that can only select the most relevant data, which has shown great potential on imbalanced big data problems. Castro et al. [62] proposed a DE algorithm to generate new data to obtain a more balanced dataset, where experiments showed that the CNN with balanced data can obtain both accuracy and recall over 90% on diagnosis problems for the six most common skin diseases.

In contrast to the data-level approach, the algorithm-level approach aims to balance the influence of imbalanced data via proper strategies and mechanisms that are designed in the algorithms. For example, Zhang et al. [63] adopted a cost-sensitive learning strategy and proposed an adaptive DE algorithm to optimize the cost of data misclassification, which was then used to train the DBN. Experiments show that this method can optimize the appropriate cost of misclassification of different categories of data without prior knowledge, and can be applied to two- and multi-class imbalanced problems. Feng et al. [64] proposed a binary ant lion algorithm to optimize the cost weights in the cost-sensitive learning strategy and to extract more suitable features to enhance the classification performance on imbalanced data. Which had shown effectiveness on 11 imbalanced datasets.

3.3. Task-specific data pre-processing

As different application scenarios may have different data pre-processing needs for obtaining better results, it would be a good idea to design appropriate EC methods to optimally pre-process the data to make the data more suitable for the requirements of specific tasks. For example, in order to obtain richer edge detection information, Zheng et al. [65] developed an improved DE algorithm to pre-process the input data of the generator in a GAN, where the output of the discriminator in the GAN is used as the fitness evaluation for the algorithm. By doing so, the proposed DE algorithm can optimize and improve the input data to be more suitable for the generator according to the discriminator output. Experimental results show that this method can quickly and efficiently improve the quality of edge detection by pre-processing the input data. To obtain better multi-view features, Liang et al. [66] proposed an EA to exact the suitable features from multi-view data and the best combination of commonly-used data fusion operators (e.g., addition and concatenation) to be performed on the selected feature data, which can generate better training and test dataset for the chemical structure recognition problem. To better study the human–computer interaction evolution in the artist-critic paradigm, Soderlund et al. [67] modeled the picture data generation problem as a constrained optimization problem and used a hierarchical EA to solve the problem, which can produce useful image data to challenge the critics based on DNN. In addition, Jiang et al. [68] proposed an improved GA to pre-process the data via feature selection, and then trained the CNN with the processed data for forecasting the outpatient demand, which can achieve good prediction results in practical applications.

3.4. Discussion

From the above review, it can be seen that there are mainly two directions for using EC to improve DL via enhancing the data processing procedure. The first direction is directly using EC to process the data, while the second is using EC to optimize the data processing methods. Due to the optimization ability of EC, processing the data directly by EC can potentially have satisfied performance. However, if the data are large-scale, using EC to optimize the data processing methods and then adopting the obtained EC-optimized processing methods to process the data can be more efficient.

4. EC for model search

When solving learning tasks, there may be not enough prior knowledge and experience to guide the design and selection of DL models. Instead, tedious trial-and-test efforts are required to find a suitable DL model. Therefore, how to automatically search for suitable models is very important for solving real-world application tasks, which has attracted dramatic attention in recent years. As finding a suitable model can be regarded as an optimization problem, many EDL works attempt to search and evolve suitable models via EC methods and have obtained promising results. In fact, there have been some surveys specially conducted on EDL for model search [44–46], where some common categorization methods are presented in Fig. 6, including categorizations based on the implementation of EC, the types of EC, and the kinds of targeted models. Different from these categorizations, this part reviews the EC methods for model search according to their targeted optimization problems, including architecture optimization, hyperparameter optimization, and multi-objective model search. For the purpose of better representation, the three categories and the corresponding issues are illustrated in Fig. 7, which will be discussed in the following contents.

4.1. Model architecture optimization

Architecture plays a vastly important role in model performance. For example, more difficult and complex classification problems may require deeper models with denser topology among layers. Therefore, various EC algorithms are adopted for optimizing the model architecture, so as to obtain more suitable architecture efficiently. Among the architecture optimization methods, most works are designed for DNNs, which are often referred as neural architecture search (NAS) methods [69]. As there are extensive works on NAS, the following contents review some representative methods. For example, a classical NAS algorithm is the large-scale evolution algorithm proposed by Real et al. [70] for architecture optimization, where the mutation operator can insert, modify, and remove some structures and topologies in the DNNs. Experimental studies suggest that the proposed algorithm can evolve better DNNs automatically according to the dataset without manual participation. Based on the idea of large-scale evolution, Real et al. [71] further modified the tournament selection operation of EA, and added the individual age (i.e. the generation numbers of survival) as a factor for individual selection. The improved algorithm can automatically and efficiently search for a better model than the existing manually-design models at that time. Considering the encoding level of DNN architecture, Saltori et al. [72] designed a cell-based encoding method and then proposed an EA to evolve the cell-based DNN architecture, while Lorenzo et al. [73] studied the topology-based encoding method and developed a memetic algorithm to optimize the topology-based DNN architecture. Song et al. [74] proposed an improved evolutionary GAN by combining different mutation operators to automatically evolve the required GAN; Hadjiivanov et al. [75] extended crossover and mutation operators and combined them to better evolve CNNs; Wang et al. [76] proposed to use covariance matrix adaptive ES to optimize the plasticity of the neural network, where the obtained model had better learning ability than the existing models; Xie et al. [77] designed an encoding method to represent network structures and defined standard genetic operations for better optimization; Chen et al. [78] proposed a GA with variable-length gene encoding strategy to search for the optimal architecture of AE; Zhang et al. [79] combined reinforced learning, I-Ching divination EA, and variable-architecture encoding strategy for efficient NAS and obtained superiority results; and Martín et al. [80] proposed a new EA to evolve DNNs to maximize the classification accuracy and maintain a sequence of layers. Moreover, as GP can evolve an appropriate model structure without relying on prior knowledge, Bi et al. [81] proposed a novel GP to automatically search for the best depth tree structure for feature learning and classification tasks. By combining the flexible depth variable coding method and the evolution operator associated with image

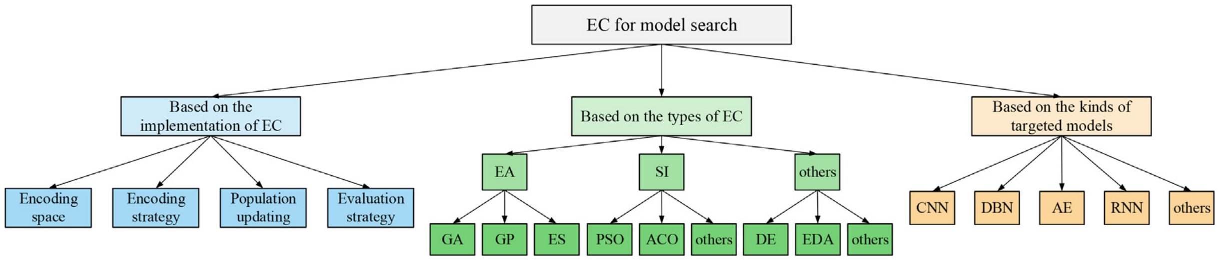

Fig. 6. Common categories of EC for searching DL models in existing works, which are categorized based on (a) the implementation of EC; (b) the types of EC; and (c) the kinds of targeted models.

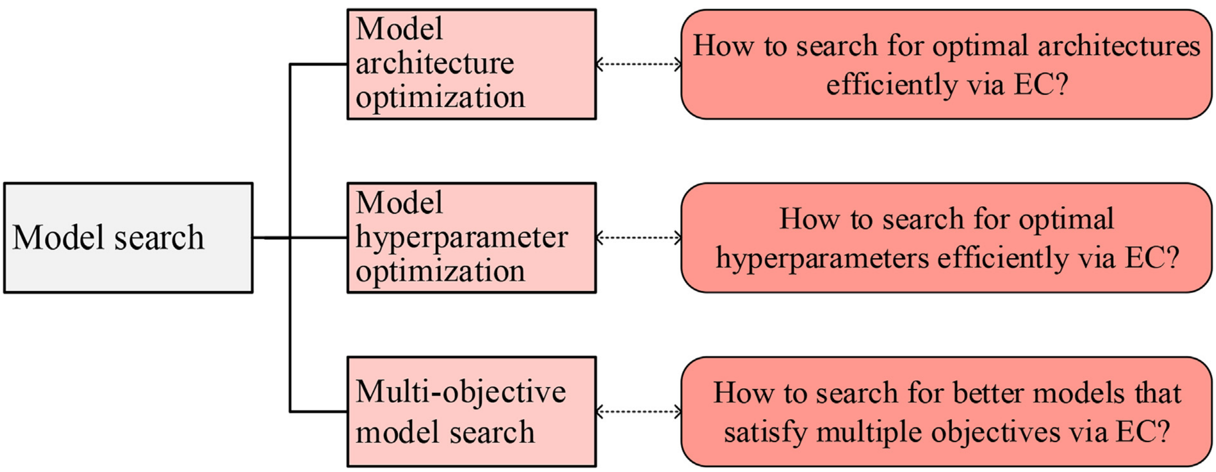

Fig. 7. Categories and corresponding issues of EDL in the model search procedure.

information, this method can evolve an efficient depth tree model. In addition, the GP combined with convolution operator has also been proved to be efficient in evolving an appropriate deep model to solve the image classification problem [82].

4.2. Model hyperparameter optimization

Besides the architecture, hyperparameter configuration is also a significant factor to influence the model performance, where common hyperparameters of some DL models are given in Table 2. Therefore, EC algorithms have also been researched for optimizing the hyperparameters configuration.

Sun et al. [83] proposed a CNN structure evolution method based on residual blocks and GA, where each block can have different configurations and the total number of blocks is also evolved based on a variable-length encoding scheme. The experiment shows that compared with other manually designed and automated model search methods, this method has better performance and lower computational overhead. Similarly, considering the advantages of residual dense blocks, Song et al. [84] proposed an improved EA to optimize the block types in the network and the hyperparameters of each block. Also based on the network block, Viswambaran et al. [85] proposed a GA to optimize the hyperparameters of the block-based RNN for the human activity recognition application. Differently, Kim et al. [86] considered the hyperparameter optimization of the layer-based DBN and proposed a novel PSO algorithm to solve the problem, which had shown efficiency in highly class imbalance classification. Also for DBN, Liu et al. [87] adopted a randomly occurring distributedly delayed PSO algorithm to search for the optimal hyperparameters, which obtained the DBN with better classification accuracy than manually-designed DBNs. Moreover, the hyperparameters optimization of the combination of multiple models has also been considered. According to the characteristics of the energy consumption prediction problem, Kim et al. [88] tried to combine the advantages of CNN and LSTM to generate a new model, and then used a PSO to automatically search for the optimal hyperparameters of the newly-generated model, where the obtained model has achieved good results on the public household electricity data set. Given that there are three types of LSTM implementation structures with different advantages, namely unidirectional, bidirectional, and cascaded structure, Viswambaran et al. [89] designed a general GA to optimize models of three connection structures, which can obtain models with better results than other methods from manual design. In addition, Sun et al. [90] conducted 192 independent comparative experiments with different settings for hyperparameter optimizations of AE between PSO algorithm and other methods such as grid search. The experiments show that PSO can achieve the same classification accuracy as other optimization methods while saving 10 to 100 times of computational overhead. Aiming at robust multimodal representation learning, Huang et al. [91] adopted the EA to efficiently select the hyperparameters for the GAN model with multimodal inputs. Furthermore, Li et al. [92] considered the hyperparameter optimization in CNN as a mixed-variable optimization problem and proposed a novel surrogate-assisted hybrid-model EDA (SHEDA) to solve the problem by efficiently optimizing the mixed-variables via an adaptive hybrid-model learning strategy. The superiority of the proposed SHEDA algorithm is verified not only on both CIFAR10- and CIFAR100-class image classification problems, but also on an aortic dissection diagnosis application with patient data from a real-world hospital. Besides, other EC methods like GP [93] have also been put forward to optimization the hyperparameters and obtain better models.

Table 2
Common Hyperparameters of DL Models for Optimization.

Model	Common hyperparameters		
	Related layer	Containing hyperparameters	Included variable types
CNN [82,83,92,93]	convolutional layer	Number of convolutional layers, kernel number, filter size (width and height), stride size (width and height), feature map size, convolution type, batch normalization type, activation function type	Integer and category
	pooling layer	Filter size (width and height), stride size (width and height), pooling type,	Integer and category
	Fully-connected layer	number of neurons, activation function type, the dropout rate	Integer, category, and continuous
DBN [9,86,87]	hidden layer	Number of hidden layers, neurons per layer, activation function type, parameters for pre-training and fine-tuning, learning rate	Integer and category
RNN [88,89]	hidden layer	Number of hidden layers, neurons per layer, number of time slot	Integer
AE [12,90]	hidden layer	Number of hidden layers, neurons per layer	Integer
GAN [91]	layers in generator and discriminator	Number of iterations of each training epoch for generator and discriminator, the parameter in the loss function of generator and discriminator	Integer and real number

4.3. Multi-objective model search

When searching for a suitable model, different issues and requirements should be considered, such as the model accuracy, computational cost, model size, and the number of model parameters. Therefore, many researches use evolutionary multi-objective optimization paradigm for better model search. For example, Cao et al. [94] considered prediction accuracy and model simplicity as two optimization objectives and proposed a distributed parallel multi-objective EA to solve the problem. Lu et al. [95] proposed an EA for searching the optimal architectures under multiple objectives, including prediction performance and floating-point operations. Moreover, Lu et al. [96] integrated many-objective evolutionary search procedures and a transfer learning method for model search, which was shown to be effective on diverse image classification tasks. Inspired by the sparse connection structure of neurons in the human brain, Liu et al. [97] modeled the model search as a multi-objective hierarchical structure learning problem and solved it by a grouping-based multi-objective EA. The proposed method is applied to a variety of DL models and obtained good results. Yang et al. [98] regarded the model search as a multi-objective optimization problem, and proposed a novel non-dominated sorting method to select candidate solutions. The experimental results show that this method can find a series of networks with better classification accuracy but fewer parameters than those of existing models. Considering that deep models are vulnerable to adversarial examples, Liu et al. [99] proposed a multi-objective EA to obtain a robust model against five types of adversarial attacks, which has obtained promising classification accuracy under different adversarial attacks. Wen et al. [100] treated the model search for the source task of transfer learning as a multi-objective optimization problem and used a modern multiobjective optimization approach to solve the problem.

4.4. Discussion

This part provides a discussion on the EC-based model search methods. For this, Table 3 compares some EC-based model search methods with distinct characteristics, which provides three observations. First, the targeted models of most compared methods are CNNs. This may be due to that the image processing applications are almost the most popular DL applications, while CNNs have a great ability for learning features and knowledge from image data. As there are various DL models, it is suggested that search methods for other models like GAN, instead of just CNN, can be further researched, so as to broaden the researches about EDL. Second, most EC algorithms in Table 3 are EA and GA, while other powerful EC algorithms like PSO, DE, and EDA are less considered. This may be due to that the variable types of the hyperparameters include not only continuous, but also discrete and category, and therefore EC algorithms that can deal with different types of variables are in great need (like the GA and SHEDA [92]). Third, when searching for different types of models, architecture and hyperparameters are usually considered, which suggests that both the architecture and hyperparameters are important for obtaining suitable models.

5. EC for model training

Model training aims to find the global optimal model parameters based on training data, which can be regarded as an optimization problem as

Table 3
Comparisons of Some EC-based Model Search Methods with Distinct Characteristics.

Method	Optimizations problem	Variable types	Encoding way	Type of targeted model	Type of EC
Regularized EA [72]	Architecture optimization	discrete	Cell-based	CNN	EA
MA [73]	Architecture optimization	discrete	Topology-based	CNN	Memetic Evolution
DRNN [85]	Hyperparameter optimization	discrete	Block-based	RNN	GA
PSO-DBN [86]	Hyperparameter optimization	discrete	Layer-based	DBN	PSO
PSO-HO [90]	Hyperparameter optimization	Continuous, discrete	Layer-based	AE	PSO
SHEDA [92]	Hyperparameter optimization	Continuous, discrete, category	Block-based mixed-variable	CNN	EDA
NSGANet variants [95]	Architecture and multi-objective optimization	discrete	Block-based	CNN	GA
NAT [96]	Hyperparameter and multi-objective optimization	discrete	Layer-based	CNN	Multi-objective EA

$$\underset{\theta_M}{\text{argmax}}\, E_{train}(M, \theta_M, D_{train}) \tag{3}$$

where θ_M is the model parameters of model M, and D_{train} is the training datasets, which can be referred to Section 2.2. However, the model training procedure is often computationally expensive due to a large amount of data and the heavy computations for updating a large number of model parameters. As a result, the model training can be regarded as an expensive optimization problem with the model parameters as corresponding variables. As EC is an efficient tool for complex expensive optimization problems [101], many EDL methods have been researched to achieve better model training. In this part, EC methods for better model training are categorized into three parts, where the three parts and the related issues are shown in Fig. 8, including model parameter optimization, model pre-training optimization, and training cost reduction.

5.1. Model parameter optimization

The model training can be viewed as a model parameter optimization problem. As an effective and efficient tool for complex problem optimization, various researches about EC have been conducted to optimize model parameters during the training process. Paulin et al. [14] used various EC algorithms, e.g., GA and PSO, to optimize the weight parameters of the constrained Boltzmann machine. Experiments on speech steganography analysis show that EC algorithms can train better constrained Boltzmann machine models than traditional training algorithms. Gong et al. [102] combined the cooperative coevolution method and backward propagation algorithm to better optimize the DNN parameters, which had achieved great results. Long et al. [103] proposed a hybrid EC algorithm based on an improved PSO and local search to optimize the parameters of the deep echo state network. Compared with the models optimized by other methods, the model optimized by the hybrid PSO algorithm has higher stability and accuracy in solving intelligent fault diagnosis problems. Similarly, Deng et al. [104] proposed an improved quantum heuristic DE algorithm to optimize the weights of all nodes in a DBN. By integrating adaptive quantum crossover and mutation operations, the proposed algorithm can avoid premature convergence and improve the global search ability, which can optimize the classification performance of the DBN in the experimental studies. Moreover, Cui et al. [105] used the EC method to jointly optimize the activation function, weight, and offset of heterogeneous DNNs. This method can train the deep network faster and better than the gradient-based optimization algorithm. In addition, Chen et al. [106] decomposed the training of proposed a cooperative dual evolution-based EA to decompose the parameter training of GAN into two subproblems (i.e., generation and discrimination), and proposed a cooperative dual evolution-based EA to solve the subproblems together, which can address the adversarial optimization difficulties in training GAN.

5.2. Model pre-training optimization

Model pre-training is one of the important technologies in DL applications, which can speed up the later training process targeted at the related training data and can avoid problems caused by the random initialization of weights. Sun et al. [107] proposed an improved GA to together optimize the structure and initial weight parameters in CNN. As there are a large number of nodes in CNN, the parameter values of random initialization may lead to the final convergence of the network to the local optimum. Therefore, the algorithm proposed by Sun et al. [107] builds the probability model based on the mean and variance of the initial weight parameters of each network layer, and then evolves the mean and variance of the probability model. After this, the weight parameters resampling from the probability model can be used as the pre-training weights of network nodes. Experimental results show that this method is more efficient than the commonly-used weight initialization methods. Moreover, Kenny et al. [108] also proposed an improved PSO with the problem decomposition strategy to pre-train the large-scale DNN, which can get better results than other commonly-used pre-training methods in the experiment.

5.3. Training cost reduction

In DL research and application, the model training is very computationally expensive and has a large time cost. Therefore, some researches made attempts to use more efficient EC methods, so as to accelerate the model search process and reduce time cost. The EC-based methods that can reduce training time can be divided into two categories, reducing the number of training required in the training process and reducing the time required for each training.

In order to reduce the number of training, Yang et al. [98] adopted the weight sharing inheritance mechanism in EC, so that the weight of each newly-evolved individual network can inherit quickly, which can reduce the unnecessary training times and accelerate the running speed of the algorithm. Sun et al. [109] proposed a surrogate-assisted evolutionary DL method, where the per-

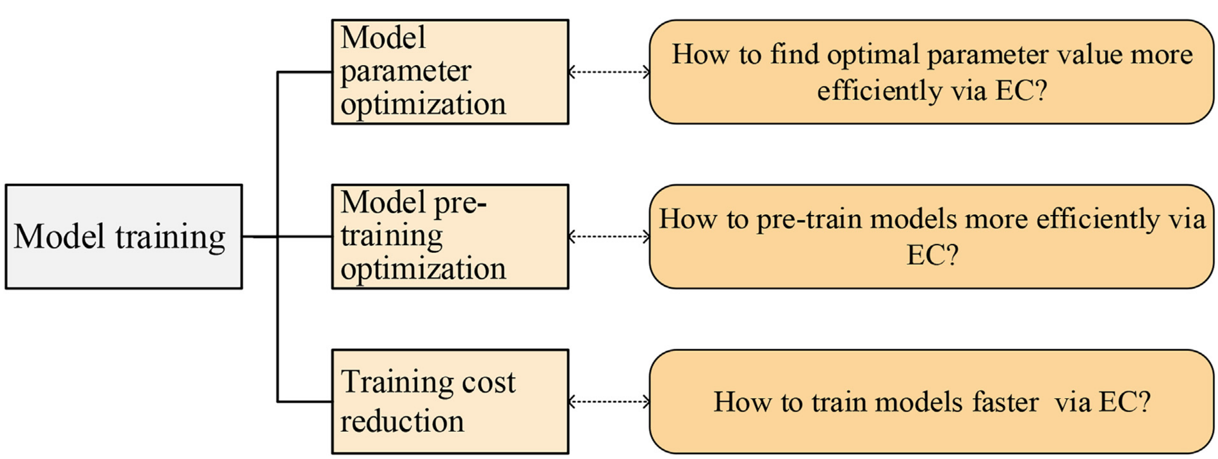

Fig. 8. Categories and corresponding issues of EDL in the model training procedure.

formance of the DL model is obtained by the prediction from an end-to-end random forest model instead of the expensive training procedure. Therefore, this method can reduce unnecessary but time-consuming training procedures and accelerate the overall search process. Similarly, Zhou et al. [110] used the Gaussian process model as a surrogate model instead of network training to evaluate candidate models obtained by EC, where experimental results show that this method is more efficient than the traditional method.

In order to reduce the time cost in each training step, the recent SHEDA proposed by Li et al. [92] adopts a surrogate-assisted multi-level evaluation method that integrates two levels of evaluation to strike a better balance between optimization accuracy and computational cost, which has shown great efficiency when compared with other methods. Also using the surrogate-assisted algorithm, Zhang et al. [111] proposed a nonstationary kernel- based Gaussian surrogate model, which can help the evolutionary strategy efficiently find the optimal model for lung nodule classification. Furthermore, Wang et al. [112] appropriately sampled training data and then used the sampled data to train the network. As the amount of sampled data is much smaller than that of all training data, the proposed sampled method can effectively reduce the computational burden and accelerate the training process. Zhu et al. [113] proposed a multi-objective evolutionary federated learning method to search the neural network structure in decentralized scenarios, which can minimize the cost of parameter communication between multiple nodes and improve the accuracy of the searched model. In order to improve the computational efficiency, Irwin-Harris et al. [114] used an EA with a graph-based encoding strategy to optimize the CNN and sampled 10% of the training data for training during the training process, greatly reducing the time consumption required for each training. In recent years, due to the development of distributed EC [115], improving the global search ability and reducing the running time of EC through parallel and distributed technology has become a mainstream research direction [116–118]. In this regard, Guo et al. [119] proposed a distributed PSO (DPSO) algorithm based on multiple GPU resources to accelerate the optimization of hyper-meters in CNN, which greatly shortened the training time in EDL. Moreover, Moriya et al. [120] used the improved multi-objective covariance matrix adaptive EA to optimize the DNN, where distributed technology and cloud computing resources were used to train the model in parallel. The experimental results show that this method has the advantages of high calculation efficiency and prediction accuracy.

5.4. Discussion

This part discusses the EC methods for model training in DL, where some representative EC-based model training methods are compared in Table 4. As can be seen, the proposed methods are mainly for parameter optimization or pre-training optimization (but not both). This is may be due to that the aims of parameter optimization and pre-training optimization are different or may contradict with each other, e.g., the parameter optimization is for the optimal parameter for the current classification task, while the pre-training optimization needs to avoid the local optimal parameter, respectively). Therefore, the parameter and pre-training optimization might be handled together via novel evolutionary optimization paradigm, like multi-objective optimization [121,122], multimodal optimization [123,124], and multi-task optimization [125]. In addition, time cost reduction techniques have not been researched adequately with complex parameter and pre-training optimization problems, which are worth further investigation.

6. EC for model evaluation and utilization

As a model may perform well on some scenarios but perform poorly in other scenarios, model evaluation and utilization are significant for the application performance of DL. Therefore, some EDL methods proposed to use EC to improve the evaluation and utilization of DL models. As shown in Fig. 9, this part reviews such related EDL methods in three categories, i.e., model robustness evaluation, model ensemble, and model pruning.

6.1. Model robustness evaluation

As the robustness of the DL model is an important factor to be considered before practical application and deployment, how to evaluate the model robustness has become a hot research topic in DL [126]. In order to efficiently measure the robustness of DL models, Su et al. [127] proposed a novel DE-based method for generating one-pixel adversarial perturbations data to attack the DL models, so as to evaluate the vulnerability and robustness of the trained models. Similarly, Vidnerová et al. [128] proposed a novel

Table 4
Comparisons of Some Representative EC-based Model Training Methods.

Method	Parameter optimization	Pretraining optimization	Time cost reduction	Type of targeted model	Type of EC
SHEDA [92]	–	–	Surrogate-assisted multi-level evaluation method	CNN	EDA
BPCC [102]	cooperative coevolution with backpropagation	–	–	CNN	DE
CSOLS [103]	Novel encoding and layer-wise optimization strategy	–	–	Deep echo state networks	PSO
MSIQDE [104]	Improved DE for parameter optimization	–	–	DBN	DE
CDE-GAN [106]	Cooperative dual-evolution EA for decomposition-based parameter optimization	–	–	GAN	EA
EvoCNN [107]	–	Optimization for initial weights	Efficient fitness evaluation	CNN	EA
SdEN [108]	–	divide-and-conquer technique for pretraining optimization	–	DNN	PSO
E2EPP [109]	–	–	Surrogate-assisted performance prediction	CNN	EA
DPSO [119]	–	–	Distributed computation with multiple GPU resources	CNN	PSO

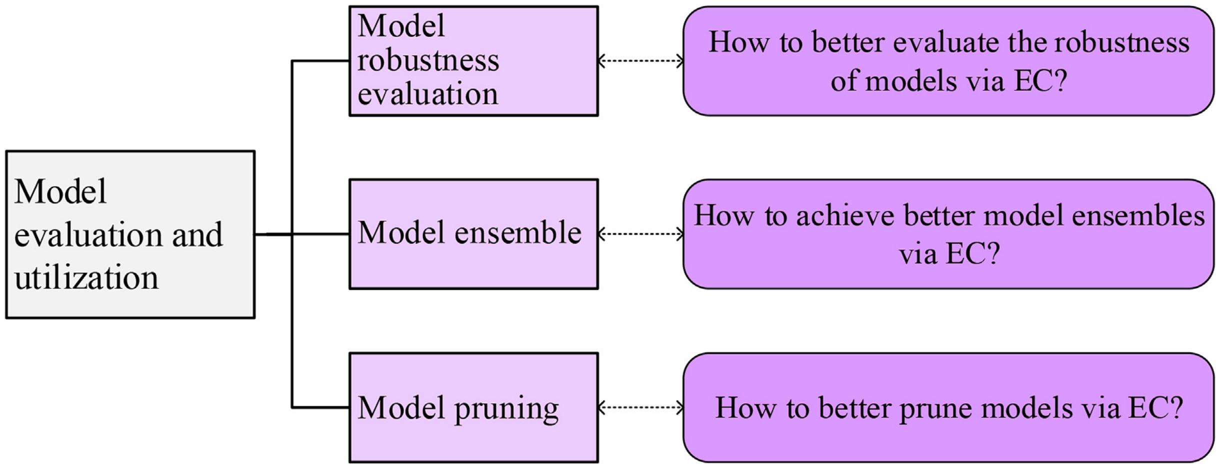

Fig. 9. Categories and corresponding issues of using EDLs in the model evaluation and utilization procedure.

EA to generate adversarial samples for testing and evaluating the vulnerability of DNN classifiers. In addition, Deng et al. [129] proposed a multi-objective EA to optimize both the imperceptibility and attack capability of data for evaluating the robustness of DL models, where the experimental studies had shown the efficiency of the proposed algorithm.

6.2. Model ensemble

Ensemble learning is a very effective method in machine learning. By combining multiple models, ensemble learning can obtain better performance (e.g., higher accuracy, higher robustness, and lower uncertainty) than a single model. With this concern, Gong et al. [130] proposed to use a multimodal DE algorithm to optimize the parameters of multiple networks at the same time, so as to obtain multiple networks with different parameters but similar performance for obtaining model ensemble. Moreover, Zhang et al. [131] used the grouping-based multi-objective EA to search for a series of DBNs whose model accuracy and model diversity are not dominated by each other, and then combined these trained models as the final ensemble model. After this, the weights of the output of different networks in the ensemble model are further optimized by a single-objective DE algorithm. Experimental results show that this method has superior performance in fault prediction and health management applications such as residual useful life estimation. Similarly, the DBN [132] using the multi-objective evolutionary ensemble method has also achieved good results in the application of fault diagnosis. Fielding et al. [133] selected the network structure represented by the personal best solution of each particle in PSO as the basic model for the ensemble. As basic models are the optima found by different particles, these basic models can have a certain diversity and high accuracy, resulting in the superior performance of the final ensemble model.

6.3. Model pruning

Although a larger and deeper model may bring better performance, it will also bring a rapid increase in computational overhead, storage requirements, and energy consumption, which limits the application scenarios of the model. Therefore, it is a very important research problem that how to compress and prune the existing models so that they can be deployed to various scenarios with limited resources, energy, and computational ability (e.g., mobile phones and edge device). For model pruning, Zhou et al. [134] proposed the evolutionary compression method to efficiently find a suitable deep network structure, and then used the proposed pruning operator to prune the model. This method has achieved good results in the segmentation of retinal blood vessels and neuronal membranes. Shu et al. [135] developed a novel co-evolutionary approach for reducing the memory usage and floating-point operations per second of the model simultaneously, which has shown efficiency in the experiments. Moreover, Wu et al. [136] developed a multi-objective neural network pruning problem with two conflicting objectives (i.e., the accuracy and sparse degree), and proposed a multi-objective PSO algorithm to solve the problem. The experimental results show that the proposed algorithm can prune more than 80% of the weight of commonly-used models, while not significantly affecting the classification accuracy of the model, which shows the potential of compression and pruning optimization.

6.4. Discussion

This part discusses EC methods for model evaluation and utilization, where some representative EC-based methods are compared in Table 5. As shown in Table 5, almost all EC-based methods are designed for CNNs, and methods for other DL models might merit further investigation. Moreover, all of the methods in Table 5 formulate the model evaluation and utilization optimization problems as single-objective or multi-objective optimization problems, which indicate that more model evaluation and utilization characteristics, such as large-scale, multi-modal, and constrained, might be considered to design better methods. In addition, more types of EC algorithms, e.g., ACO [137,138], have not been sufficiently investigated so far, which may deserve more attention.

7. Potential future research directions

As mentioned in the Section 2–6, many researchers have been successfully realized and made good progresses on the idea of using EC to improve DL. Moreover, we can summary the major technical difficulties when using EC for optimizing DL from the above survey and discussions in this section. Table 6 provides the common technical difficulties when using EC for enhancing DL in data processing, model search, model training, and model evaluation and utilization, respectively. As can be seen, solution encoding, large-scale search space, and expensive time and computational cost are common difficulties in different EDL procedures. In fact, these difficulties become more obvious as the model goes deeper

Table 5
Comparisons of Some EC-based Model Evaluation and Utilization Methods with Distinct Characteristics.

Method	For model evaluation	For model utilization	Type of targeted model	Type of EC
One-pixel attack [127]	Adversarial sample optimization to evaluate model robustness and vulnerability	–	CNN	DE
Black box attacks [128]	Adversarial sample optimization to evaluate model robustness and vulnerability	–	CNN	GA
MOEA-APGA [129]	Multi-objective adversarial sample optimization	–	CNN	Multi-objective GA
multi-objective ensemble [132]	–	Multi-objective optimization for ensemble weight	DBN	Multi-objective EA
SOBAE [133]	–	Ensemble of best networks found by particles	CNN	PSO
ECDNN [134]	–	Model pruning	CNN	EA
MOPSO [136]	–	Multi-objective network pruning	CNN	PSO

Table 6
Common Technical Difficulties in Different Procedures of EDL.

EDL Procedure	Common technical difficulties
EC for data processing	• Solution encoding • Large-scale search space • Expensive time and computational cost
EC for model search	• Solution encoding • Large-scale search space • Mixed variable optimization • Interpretability of the found model • Expensive time and computational cost
EC for model training	• Solution encoding • Large-scale search space • Expensive time and computational cost
EC for model evaluation and utilization	• Solution encoding • Large-scale search space • Expensive time and computational cost

and the data becomes bigger. Moreover, as shown in Table 6, EC for model search often encounters more difficulties than other procedures, e.g., the mixed variable optimization for hyperparameter optimization. This indicates that more issues and problems need to be addressed better when designing EC algorithms for model search. Therefore, it is no wonder that most of existing EDL researches cover the procedure of EC for model search.

Moreover, we find that the research in this field is not systematic enough, and there are still many problems that are worth further investigation and exploration in the future. Therefore, this section will also discuss some prospective research directions to inspire more thorough and in-depth research into EDL, which can also be significant research topics in EDL in the near future. To be more specific, the following contents will consider and discuss five potential future directions from three levels, i.e., resource-algorithm-application levels, as illustrated in Fig. 10.

7.1. Benchmark, toolbox, and platform for EDL

Although many EDL methods have been proposed, it is not easy to compare existing methods under the same conditions, which pose great challenges to the analysis and investigation of different EDL methods. This is mainly due to that most comparisons do not adopt the same experimental setting and datasets and different EDL methods usually have distinct encoding schemes, search space, and training tricks including data processing methods. For example, when considering two EDL methods with different encoding schemes and search space for architecture search, the best model found by one method may never be generated by the

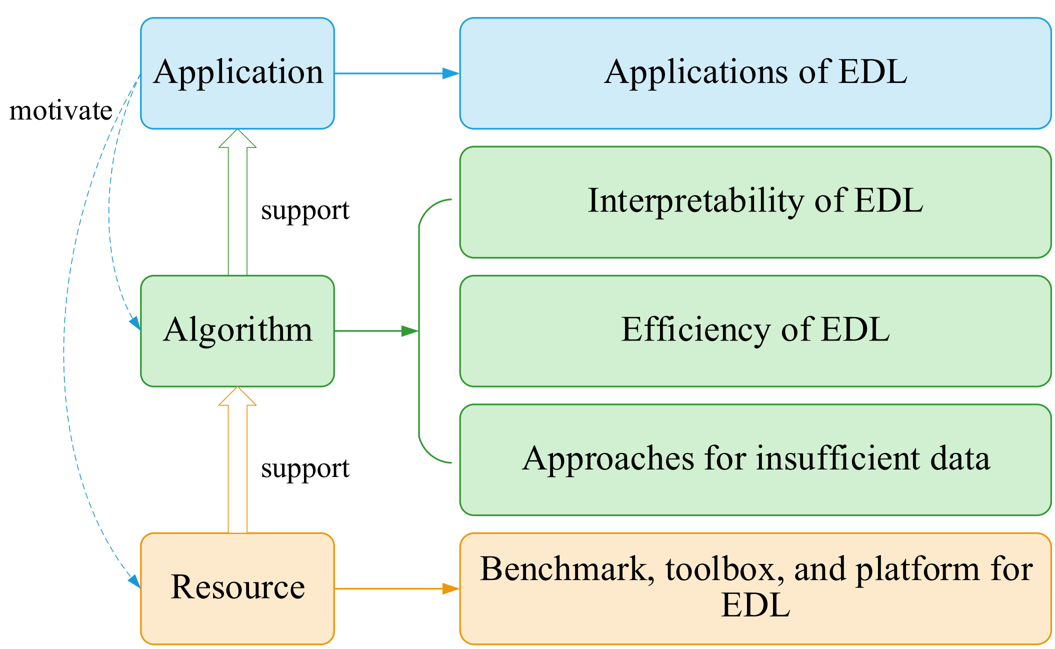

Fig. 10. Some potential future research directions of EDL.

other method. Moreover, some widely-used search spaces of network architecture are poorly constructed and sometimes random search in such search spaces can perform better than other search methods [139], which are not suitable for investigating the search efficiency of different EDL methods. Recently, some researches have made attempts to propose EDL datasets to evaluate the search efficiency of EDL methods. For example, Duan et al. [140] proposed a benchmark dataset named TransNAS-Bench-101 across seven tasks for investigating the transferability and generalizability of architecture search methods. In this dataset, all possible DNN architectures in the pre-defined search space of each task have been fully trained in advance. With this dataset, different EDL methods can employ the same search space for the corresponding task to guarantee the fairness of comparison. Furthermore, there is no need to conduct the time-consuming training process during architecture evolution, because all possible DNN architectures have been pre-trained. Therefore, such a benchmark reduces the time and computational cost and allows the researchers to focus on designing and developing more effective EDL methods. However, there is still a need for benchmarks that include more complex EDL problems with various difficulties and characteristics, such as the large-scale [141], and dynamic [142], and multimodal characteristics [143,144]. Furthermore, benchmarks for other issues, e.g., hyperparameter optimization, model ensemble, model pruning, are also worthy of study.

Besides benchmark, it is also highly desirable to develop an open EDL toolbox and platform for easy usage and integration of new EDL methods. This would facilitate the empirical studies of EDL methods, and lower the barriers faced by new researchers and practitioners in related fields. Although developing an EDL toolbox and platform that contain representative EDL methods and benchmarks can be very challenging, it has substantial academic and practical significance and it is expected by researchers and users of various fields.

7.2. Approaches for insufficient data

EDL often requires a substantial amount of labeled data to train and validate models. However, in real-world applications, the amount of labeled data is frequently limited, which poses great challenges to EDL. Moreover, available unlabeled data can be of great amount but they are usually wasted or not well utilized in real-world problems. Some approaches for overcoming these challenges have been researched for DL recently, such as meta-learning [145], unsupervised learning [146], and self-supervised learning [147], which have shown promising results. However, these works are not yet sufficient for real-world applications nowadays. Therefore, how to better integrate EDL and these approaches deserve further exploration.

7.3. Efficiency of EDL

As the fitness evaluation of EDL (i.e., training and validating the model) is typically computationally expensive, especially for big data application problems, it is vitally essential to increase the efficiency of EDL, i.e., reduce computational cost. Similar to the contents discussed in Section 5.3, there can be two ways to accomplish this goal: 1) minimize the overall number of required fitness evaluations (i.e., model training) as much as possible, and 2) reduce the average cost of assessing a fitness evaluation. The first way indicates to propose and use more advanced EC to find the global optimum more efficiently, so that the required number of fitness evaluations can be cut down. That is, the population size of the EC or/and the generation of the evolution are reduced. The second way requires efficient and effective evaluation methods to substitute the original expensive evaluations, so as to decrease the average cost for each evaluation. These two approaches can be commonly-seen in EC for complex optimization problems, especially expensive optimization problems [148–150], data-driven optimization problems [151–154], and large-scale optimization problems [155–159]. In fact, the EDL is similar to these optimization problems in that their fitness evaluations are computationally expensive. Therefore, the EC-based techniques for these optimization problems can be also referenced to help improve the efficiency of EDL.

Moreover, when considering the efficiency of EDL, it is also of great need to study the tradeoff between computational complexity and model accuracy in EDL, because both the DL models and EC algorithms have high-computational complexity. As the computational complexity and model accuracy are two different metrics, some researches considered the trade-off as a multi-objective optimization problem. For example, Lu et al. [95] proposed to search the optimal architectures under multiple objectives, including prediction performance (for model accuracy) and floating-point operations (for computational complexity). Different from the multi-objective approach, Sun et al. [109] compared the consumed GPU days of different EDL methods when the accuracy of found model satisfies a pre-defined condition, so as to compare the efficiency of different EDL methods. However, the study about the tradeoff between computational complexity and model accuracy in EDL is still insufficient, which requires further studies.

7.4. Interpretability of EDL

As the EDL becomes more complex, it will be more difficult to clarify why and how the EDL can get better results. Moreover, deep models are usually uninterpretable due to the enormous number of learned features, which is a big barrier to their applications in real-world problems. Although some studies [160] have used EC to automatically design interpretable NNs, these works only aim at shallow NNs and the number of the generated features is relatively small. Therefore, more studies towards EDL with high performance and improved interpretability are essential.

Moreover, the evolution process can be more interpretable if the EDL can be integrated with some experiences and knowledge of experts. Generally speaking, if a manually designed model is recommended by experts or is possibly suitable for the new target real-world application, this model can be considered as a candidate solution in the initialization procedure of EDL, and then the EDL can find better models based on this model. Therefore, the experiences and knowledge of experts can be integrated with or embedded into the EDL, so as to automatically evolve promising models more efficiently and more interpretable.

7.5. Applications of EDL

Since EDL has shown superior performance in common datasets, these efficient methods should be extended to practical application problems in the next step. Moreover, as EDL has the advantages of self-adjustment, self-learning, and self-evolution, they have the potential to be applied to various types of application problems. In addition, in the process of applying EDL to practical problems, it is possible to inspire new ideas and form a series of breakthrough methods and technologies with strong generalization ability. Therefore, applying EDL to a wide range of real-world applications can also be a promising research direction.

8. Conclusion

This paper systematically summarizes existing EDL works, i.e., the methods and technologies for optimizing DL using EC. The existing EDL works are categorized into four categories according to the four procedures in the whole lifetime of DL, namely EC for data processing, model search, model training, and model evaluation and utilization. In each category, the usage of EC is further classified according to their motivations and aims. Such hierarchical categorization helps to better understand when and how EC can be used to improve DL. At the same time, this survey reveals the reasons and ways that EC can improve DL performance in different aspects. Finally, based on the overview and classification of existing work, this paper discusses and prospects some potential research directions in the future. We hope this survey can well-organize existing works, inspire new ideas, raise more and greater attentions, and extend wider and deeper research in this emerging, exciting, and interesting field.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by the National Key Research and Development Program of China under Grant 2019YFB2102102, in part by the National Natural Science Foundations of China (NSFC) under Grant 62176094, Grant 61822602, Grant 61772207, and Grant 61873097, in part by the Key-Area Research and Development of Guangdong Province under Grant 2020B010166002, in part by the Guangdong Natural Science Foundation Research Team under Grant 2018B030312003, and in part by the National Research Foundation of Korea (NRF-2021H1D3A2A01082705).

References

[1] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444, https://doi.org/10.1038/nature14539.
[2] X. Yan, S. Hu, Y. Mao, Y. Ye, H. Yu, Deep multi-view learning methods: A review, Neurocomputing 448 (2021) 106–129, https://doi.org/10.1016/j.neucom.2021.03.090.
[3] I. Lauriola, A. Lavelli, F. Aiolli, An introduction to deep learning in natural language processing: Models, techniques, and tools, Neurocomputing 470 (2022) 443–456, https://doi.org/10.1016/j.neucom.2021.05.103.
[4] C. Li, S. Zhang, Y. Qin, E. Estupinan, A systematic review of deep transfer learning for machinery fault diagnosis, Neurocomputing 407 (2020) 121–135, https://doi.org/10.1016/j.neucom.2020.04.045.
[5] S. Jia, S. Jiang, Z. Lin, N. Li, M. Xu, S. Yu, A survey: Deep learning for hyperspectral image classification with few labeled samples, Neurocomputing 448 (2021) 179–204, https://doi.org/10.1016/j.neucom.2021.03.035.
[6] Z. Niu, G. Zhong, H. Yu, A review on the attention mechanism of deep learning, Neurocomputing 452 (2021) 48–62, https://doi.org/10.1016/j.neucom.2021.03.091.
[7] S. Kuutti, R. Bowden, Y. Jin, P. Barber, S. Fallah, A survey of deep learning applications to autonomous vehicle control, IEEE Trans. Intell. Transp. Syst. 22 (2) (2021) 712–733.
[8] Y. Sun, B. Xue, M. Zhang, G.G. Yen, J. Lv, Automatically designing CNN architectures using the genetic algorithm for image classification, IEEE Trans. Cybern. 50 (9) (2020) 3840–3854.
[9] Y. Hua, J. Guo, H. Zhao, Deep belief networks and deep learning, Proc. 2015 Int. Conf. Intell. Comput. Internet Things, ICIT 2015. (2015) 1–4. https://doi.org/10.1109/ICAIOT.2015.7111524.
[10] N. Zeng, H. Li, Y. Peng, A new deep belief network-based multi-task learning for diagnosis of Alzheimer's disease, Neural Comput. Appl. 6 (2021), https://doi.org/10.1007/s00521-021-06149-6.
[11] N. Zeng, H. Li, Z. Wang, W. Liu, S. Liu, F.E. Alsaadi, X. Liu, Deep-reinforcement-learning-based images segmentation for quantitative analysis of gold immunochromatographic strip, Neurocomputing 425 (2021) 173–180, https://doi.org/10.1016/j.neucom.2020.04.001.
[12] Y. Sun, B. Xue, M. Zhang, G.G. Yen, A particle swarm optimization-based flexible convolutional autoencoder for image classification, IEEE Trans. Neural Networks Learn. Syst. 30 (8) (2019) 2295–2309, https://doi.org/10.1109/TNNLS.2018.2881143.
[13] N. Xue, I. Triguero, G.P. Figueredo, D. Landa-Silva, Evolving deep CNN-LSTMs for inventory time series prediction, in: Proc. IEEE Congr. Evol. Comput. CEC (2019) 1517–1524. https://doi.org/10.1109/CEC.2019.8789957.
[14] C. Paulin, S.A. Selouani, E. Hervet, Speech steganalysis using evolutionary restricted Boltzmann machines, in: Proc. IEEE Congr. Evol. Comput. (2016) 4831–4838. https://doi.org/10.1109/CEC.2016.7744409.
[15] Z.H. Zhan, L. Shi, K.C. Tan, J. Zhang, A survey on evolutionary computation for complex continuous optimization, Artif. Intell. Rev. 55 (1) (2022) 59–110, https://doi.org/10.1007/s10462-021-10042-y.
[16] Z.H. Zhan, J. Zhang, Y. Lin, J.Y. Li, T. Huang, X.Q. Guo, F.F. Wei, S. Kwong, X.Y. Zhang, R. You, Matrix-based evolutionary computation, IEEE Trans. Emerg. Top. Comput. Intell. (2021), https://doi.org/10.1109/TETCI.2020.3047410.
[17] D.B. Fogel, Evolutionary Computation: Toward A New Philosophy of Machine Intelligence, IEEE Press, Piscataway, USA, 1995.
[18] J. Kennedy, R. Eberhart, Y. Shi, Swarm Intelligence, Morgan Kaufmann, San Mateo, CA, 2001.
[19] D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, Boston, MA, USA, 1989.
[20] S.C. Liu, Z.G. Chen, Z.H. Zhan, S.W. Jeon, S. Kwong, J. Zhang, Many-objective job-shop scheduling: a multiple populations for multiple objectives-based genetic algorithm approach, IEEE Trans. Cybern. (2021), https://doi.org/10.1109/TCYB.2021.3102642.
[21] H.G. Beyer, H.P. Schwefel, Evolution strategies—A comprehensive introduction, Natural Comput. 1 (2002) (2002) 43–52.
[22] J.R. Koza, Genetic Programming: On the Programming of Computers by Means of Natural Selection, MIT Press, Cambridge, MA, USA, 1992.
[23] X. Yao, Y. Liu, G. Lin, Evolutionary programming made faster, IEEE Trans. Evol. Comput. 3 (1999) 82–102, https://doi.org/10.1109/4235.771163.
[24] R. Storn, K. Price, Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces, J. Glob. Optim. 11 (1997) 341–359, https://doi.org/10.1023/A:1008202821328.
[25] Z.J. Wang, Z.H. Zhan, Y. Lin, W.J. Yu, H. Wang, S. Kwong, J. Zhang, Automatic niching differential evolution with contour prediction approach for multimodal optimization problems, IEEE Trans. Evol. Comput. 24 (1) (2020) 114–128, https://doi.org/10.1109/TEVC.2019.2910721.
[26] J.Y. Li, Z.H. Zhan, K.C. Tan, J. Zhang, A meta-knowledge transfer-based differential evolution for multitask optimization, IEEE Trans. Evol. Comput. (2021), https://doi.org/10.1109/TEVC.2021.3131236.
[27] X.F. Liu, Z.H. Zhan, Y. Lin, W.N. Chen, Y.J. Gong, T.L. Gu, H.Q. Yuan, J. Zhang, Historical and heuristic-based adaptive differential evolution, IEEE Trans. Syst. Man Cybern. Syst. 49 (12) (2019) 2623–2635, https://doi.org/10.1109/TSMC.2018.2855155.
[28] P. Larrañaga, J.A. Lozano, Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation, Kluwer Academic, Boston, MA, 2002.
[29] Z.-G. Chen, Y. Lin, Y.-J. Gong, Z.-H. Zhan, J. Zhang, Maximizing lifetime of range-adjustable wireless sensor networks: A neighborhood-based estimation of distribution algorithm, IEEE Trans. Cybern. 51 (11) (2021) 5433–5444, https://doi.org/10.1109/TCYB.2020.2977858.
[30] M. Dorigo, L.M. Gambardella, Ant colony system: A cooperative learning approach to the traveling salesman problem, IEEE Trans. Evol. Comput. 1 (1997) 53–66, https://doi.org/10.1109/4235.585892.
[31] Z.-G. Chen, Z.-H. Zhan, Y. Lin, Y.-J. Gong, T.-L. Gu, F. Zhao, H.-Q. Yuan, X. Chen, Q. Li, J. Zhang, Multiobjective cloud workflow scheduling: a multiple populations ant colony system approach, IEEE Trans. Cybern. 49 (8) (2019) 2912–2926, https://doi.org/10.1109/TCYB.2018.2832640.
[32] S.-Z. Zhou, Z.-H. Zhan, Z.-G. Chen, S. Kwong, J. Zhang, A multi-objective ant colony system algorithm for airline crew rostering problem with fairness and satisfaction, IEEE Trans. Intell. Transp. Syst. 22 (11) (2021) 6784–6798.
[33] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: Proc. Int. Conf. Neural Networks, 1995, pp. 1942–1948.
[34] X. Zhang, K.-J. Du, Z.-H. Zhan, S. Kwong, T.-L. Gu, J. Zhang, Cooperative coevolutionary bare-bones particle swarm optimization with function independent decomposition for large-scale supply chain network design with uncertainties, IEEE Trans. Cybern. 50 (10) (2020) 4454–4468, https://doi.org/10.1109/TCYB.2019.2937565.
[35] X.-F. Liu, Z.-H. Zhan, Y. Gao, J. Zhang, S. Kwong, J. Zhang, Coevolutionary particle swarm optimization with bottleneck objective learning strategy for many-objective optimization, IEEE Trans. Evol. Comput. 23 (4) (2019) 587–602, https://doi.org/10.1109/TEVC.2018.2875430.
[36] W. Liu, Z. Wang, N. Zeng, Y. Yuan, F.E. Alsaadi, X. Liu, A novel sigmoid-function-based adaptive weighted particle swarm optimizer, IEEE Trans. Cybern. 51 (2021) 1085–1093, https://doi.org/10.1007/s13042-020-01186-4.
[37] N. Zeng, Z. Wang, W. Liu, H. Zhang, K. Hone, X. Liu, A dynamic neighborhood-based switching particle swarm optimization algorithm, IEEE Trans. Cybern. (2020) 1–12, https://doi.org/10.1109/TCYB.2020.3029748.
[38] N. Zeng, D. Song, H. Li, Y. You, Y. Liu, F.E. Alsaadi, A competitive mechanism integrated multi-objective whale optimization algorithm with differential evolution, Neurocomputing 432 (2021) 170–182, https://doi.org/10.1016/j.neucom.2020.12.065.
[39] X. Yao, Evolving artificial neural networks, Proc. IEEE 87 (9) (1999) 1423–1447.
[40] J. Zhang, Z.H. Zhan, Y. Lin, N.i. Chen, Y.J. Gong, J.H. Zhong, H.S.H. Chung, Y. Li, Y. H. Shi, Evolutionary computation meets machine learning: a survey, IEEE Comput. Intell. Mag. 6 (4) (2011) 68–75, https://doi.org/10.1109/MCI.2011.942584.

[41] H. Al-Sahaf, Y. Bi, Q.i. Chen, A. Lensen, Y.i. Mei, Y. Sun, B. Tran, B. Xue, M. Zhang, A survey on evolutionary machine learning, J. R. Soc. New Zeal. 49 (2) (2019) 205–228, https://doi.org/10.1080/03036758.2019.1609052.
[42] Z.G. Chen, Z.H. Zhan, W. Shi, W. Chen, J. Zhang, When neural network computation meets evolutionary computation: a survey, Int. Symp. Neural Networks (2016) 603–612, https://doi.org/10.1007/978-3-319-40663-3.
[43] M. Li, Z. Wang, K. Li, X. Liao, K. Hone, X. Liu, Task allocation on layered multiagent systems: When evolutionary many-objective optimization meets deep Q-learning, IEEE Trans. Evol. Comput. 25 (5) (2021) 842–855, https://doi.org/10.1109/TEVC.2021.3049131.
[44] Y. Liu, Y. Sun, M. Bing Xue, G.G. Zhang, K.C.T. Yen, A survey on evolutionary neural architecture search, IEEE Trans. Neural Networks Learn. Syst. (2021) 1–11, https://doi.org/10.1109/TNNLS.2021.3100554.
[45] X. Zhou, A.K. Qin, M. Gong, K.C. Tan, A survey on evolutionary construction of deep neural networks, IEEE Trans. Evol. Comput. 25 (5) (2021) 894–912, https://doi.org/10.1109/TEVC.2021.3079985.
[46] X. Zhou, C. Tan, A survey of advances in evolutionary neural architecture search, in: Proc. IEEE Congr. Evol. Comput., 2018, pp. 950–957.
[47] A. Darwish, A.E. Hassanien, S. Das, A survey of swarm and evolutionary computing approaches for deep learning, Artif. Intell. Rev. 53 (3) (2020) 1767–1812, https://doi.org/10.1007/s10462-019-09719-2.
[48] V. Bharti, B. Biswas, K.K. Shukla, Recent trends in nature inspired computation with applications to deep learning, in: Proc. Conflu. Int. Conf. Cloud Comput. Data Sci. Eng., 2020, pp. 294–299. https://doi.org/10.1109/Confluence47617.2020.9057841.
[49] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 770–778. https://doi.org/10.1109/CVPR.2016.90.
[50] D. Silver, A. Huang, C.J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, D. Hassabis, Mastering the game of Go with deep neural networks and tree search, Nature 529 (7587) (2016) 484–489, https://doi.org/10.1038/nature16961.
[51] C. Wang, C. Xu, X. Yao, D. Tao, Evolutionary generative adversarial networks, IEEE Trans. Evol. Comput. 23 (6) (2019) 921–934, https://doi.org/10.1109/TEVC.2019.2895748.
[52] W. Liu, Z. Wang, L. Tian, S. Lauria, X. Liu, Melt pool segmentation for additive manufacturing: A generative adversarial network approach, Comput. Electr. Eng. 92 (2021) 107183, https://doi.org/10.1016/j.compeleceng.2021.107183.
[53] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, F.E. Alsaadi, A survey of deep neural network architectures and their applications, Neurocomputing 234 (2017) 11–26, https://doi.org/10.1016/j.neucom.2016.12.038.
[54] E. Papavasileiou, J. Cornelis, B. Jansen, A systematic literature review of the successors of 'neuroevolution of augmenting topologies', Evol. Comput. 29 (1) (2021) 1–73, https://doi.org/10.1162/evco_a_00282.
[55] K. Mehta, Z. Kobti, K. Pfaff, S. Fox, Data augmentation using CA evolved GANs, in: Proc. Int. Symp. Comput. Commun., 2019, pp.1087–1092. https://doi.org/10.1109/ISCC47284.2019.8969638.
[56] P. Dutta, S. Saha, A weak supervision technique with a generative model for improved gene clustering, in: Proc. IEEE Congr. Evol. Comput., 2019, pp. 2521–2528. https://doi.org/10.1109/CEC.2019.8790052.
[57] J. Ma, D.O. Afolabi, J. Ren, A. Zhen, Predicting seminal quality via imbalanced learning with evolutionary safe-level synthetic minority over-sampling technique, Cognit. Comput. 13 (4) (2021) 833–844, https://doi.org/10.1007/s12559-019-09657-9.
[58] H.L. Le, D. Landa-Silva, M. Galar, S. Garcia, I. Triguero, EUSC: A clustering-based surrogate model to accelerate evolutionary undersampling in imbalanced classification, Appl. Soft Comput. 101 (2021) 107033, https://doi.org/10.1016/j.asoc.2020.107033.
[59] S.E. Roshan, S. Asadi, Improvement of Bagging performance for classification of imbalanced datasets using evolutionary multi-objective optimization, Eng. Appl. Artif. Intell. 87 (2020) 103319, https://doi.org/10.1016/j.engappai.2019.103319.
[60] Y. Zhu, Y. Yan, Y. Zhang, Y. Zhang, EHSO: Evolutionary hybrid sampling in overlapping scenarios for imbalanced learning, Neurocomputing 417 (2020) 333–346, https://doi.org/10.1016/j.neucom.2020.08.060.
[61] I. Triguero, M. Galar, H. Bustince, F. Herrera, A first attempt on global evolutionary undersampling for imbalanced big data, in: Proc. IEEE Congr. Evol. Comput. CEC, 2017, pp. 2054–2061. https://doi.org/10.1109/CEC.2017.7969553.
[62] P.B.C. Castro, B. Krohling, A.G.C. Pacheco, R.A. Krohling, An app to detect melanoma using deep learning: An approach to handle imbalanced data based on evolutionary algorithms, Proc. Int. Jt. Conf. Neural Networks (2020) 1–7, https://doi.org/10.1109/IJCNN48605.2020.9207552.
[63] C. Zhang, K.C. Tan, H. Li, G.S. Hong, A cost-sensitive deep belief network for imbalanced classification, IEEE Trans. Neural Networks Learn. Syst. 30 (1) (2019) 109–122, https://doi.org/10.1109/TNNLS.2018.2832648.
[64] F. Feng, K.-C. Li, J. Shen, Q. Zhou, X. Yang, Using cost-sensitive learning and feature selection algorithms to improve the performance of imbalanced classification, IEEE Access 8 (2020) 69979–69996, https://doi.org/10.1109/ACCESS.2020.2987364.
[65] W. Zheng, C. Gou, L. Yan, F. Y. Wang, Differential-evolution-based generative adversarial networks for edge detection, in: Proc. Int. Conf. Comput. Vis. Work, 2019, pp. 2999–3008. https://doi.org/10.1109/ICCVW.2019.00362.
[66] X. Liang, Q. Guo, Y. Qian, W. Ding, Q. Zhang, Evolutionary deep fusion method and its application in chemical structure recognition, IEEE Trans. Evol. Comput. 25 (5) (2021) 883–893, https://doi.org/10.1109/TEVC.2021.3064943.
[67] J. Soderlund, A. Blair, Adversarial image generation using evolution and deep learning, Proc. IEEE Congr. Evol. Comput. (2018), https://doi.org/10.1109/CEC.2018.8477754.
[68] S. Jiang, K.S. Chin, L. Wang, G. Qu, K.L. Tsui, Modified genetic algorithm-based feature selection combined with pre-trained deep neural network for demand forecasting in outpatient department, Expert Syst. Appl. 82 (2017) 216–230, https://doi.org/10.1016/j.eswa.2017.04.017.
[69] T. Elsken, J.H. Metzen, F. Hutter, A survey on neural architecture search, J. Mach. Learn. Res. 20 (2019) 1–21.
[70] E. Real, S. Moore, A. Selle, S. Saxena, Y.L. Suematsu, J. Tan, Q. V. Le, A. Kurakin, Large-scale evolution of image classifiers, in: Proc. Int. Conf. Mach. Learn., 2017, pp. 4429–4446.
[71] E. Real, A. Aggarwal, Y. Huang, Q. V. Le, Regularized evolution for image classifier architecture search, in: Proc. AAAI Conf. Artif. Intell., 2019, pp. 4780–4789. https://doi.org/10.1609/aaai.v33i01.33014780.
[72] C. Saltori, S. Roy, N. Sebe, G. Iacca, Regularized evolutionary algorithm for dynamic neural topology search, Springer International Publishing (2019), https://doi.org/10.1007/978-3-030-30642-7_20.
[73] P. R. Lorenzo, J. Nalepa, Memetic evolution of deep neural networks, in: Proc. 2018 Genet. Evol. Comput. Conf., 2018, pp. 505–512. https://doi.org/10.1145/3205455.3205631.
[74] J. Song, Y. Jin, Y. Li, C. Lang, Learning structural similarity with evolutionary-GAN: A New face de-identification method, in: Proc. Int. Conf. Behav. Econ. Socio-Cultural Comput., 2019, pp. 1–8. https://doi.org/10.1109/BESC48373.2019.8962993.
[75] A. Hadjiivanov, A. Blair, Epigenetic evolution of deep convolutional models, 2019 in: Proc. IEEE Congr. Evol. Comput., 2019, pp. 1478–1486. https://doi.org/10.1109/CEC.2019.8790327.
[76] X. Wang, Y. Jin, K. Hao, Evolving local plasticity rules for synergistic learning in echo state networks, IEEE Trans. Neural Networks Learn. Syst. 31 (4) (2020) 1363–1374, https://doi.org/10.1109/TNNLS.2019.2919903.
[77] L. Xie, A. Yuille, Genetic CNN, in: Proc. IEEE Int. Conf. Comput. Vis. 2017-Octob, 2017, pp. 1388–1397. https://doi.org/10.1109/ICCV.2017.154.
[78] X. Chen, Y. Sun, M. Zhang, D. Peng, Evolving Deep convolutional variational autoencoders for image classification, IEEE Trans. Evol. Comput. 25 (5) (2021) 815–829, https://doi.org/10.1109/TEVC.2020.3047220.
[79] T. Zhang, C. Lei, Z. Zhang, X.-B. Meng, C.L.P. Chen, AS-NAS: adaptive scalable neural architecture search with reinforced evolutionary algorithm for deep learning, IEEE Trans. Evol. Comput. 25 (5) (2021) 830–841, https://doi.org/10.1109/TEVC.2021.3061466.
[80] A. Martín, R. Lara-Cabrera, F. Fuentes-Hurtado, V. Naranjo, D. Camacho, EvoDeep: A new evolutionary approach for automatic deep neural networks parametrisation, J. Parallel Distrib. Comput. 117 (2018) 180–191, https://doi.org/10.1016/j.jpdc.2017.09.006.
[81] Y. Bi, B. Xue, M. Zhang, Genetic programming with image-related operators and a flexible program structure for feature learning in image classification, IEEE Trans. Evol. Comput. 25 (1) (2021) 87–101.
[82] Y. Bi, B. Xue, M. Zhang, An evolutionary deep learning approach using genetic programming with convolution operators for image classification, in: Proc.2019 IEEE Congr. Evol. Comput., 2019, pp. 3197–3204. https://doi.org/10.1109/CEC.2019.8790151.
[83] Y. Sun, B. Xue, M. Zhang, G.G. Yen, Completely automated CNN architecture design based on blocks, IEEE Trans. Neural Networks Learn. Syst. 31 (4) (2020) 1242–1254, https://doi.org/10.1109/TNNLS.2019.2919608.
[84] D. Song, C. Xu, X. Jia, Y. Chen, C. Xu, Y. Wang, Efficient residual dense block search for image super-resolution, Proc. Thirty-Fourth AAAI Conf. Artif. Intell. (2020) 12007–12014.
[85] R.A. Viswambaran, G. Chen, B. Xue, M. Nekooei, Evolutionary design of recurrent neural network architecture for human activity recognition, in: Proc. IEEE Congr. Evol. Comput. CEC, 2019, pp. 554–561. https://doi.org/10.1109/CEC.2019.8790050.
[86] J.K. Kim, Y.S. Han, J.S. Lee, Particle swarm optimization–deep belief network–based rare class prediction model for highly class imbalance problem, Concurr. Comput. Pract. Exp. 29 (11) (2017), https://doi.org/10.1002/cpe.4128.
[87] W. Liu, Z. Wang, N. Zeng, F.E. Alsaadi, X. Liu, A PSO-based deep learning approach to classifying patients from emergency departments, Int. J. Mach. Learn. Cybern. 12 (7) (2021) 1939–1948, https://doi.org/10.1007/s13042-021-01285-w.
[88] T. Y. Kim, S.B. Cho, Particle swarm optimization-based CNN-LSTM networks for forecasting energy consumption, in: Proc. IEEE Congr. Evol. Comput., 2019, pp. 1510–1516. https://doi.org/10.1109/CEC.2019.8789968.
[89] R. A. Viswambaran, G. Chen, B. Xue, M. Nekooei, Evolutionary design of recurrent neural network architecture for human activity recognition, in Proc. IEEE Congr. Evol. Comput., 2019, pp. 554–561. https://doi.org/10.1109/CEC.2019.8790050.
[90] Y. Sun, B. Xue, M. Zhang, G. G. Yen, An experimental study on hyper-parameter optimization for stacked auto-encoders, in Proc. IEEE Congr. Evol. Comput., 2018, pp. 1–8. https://doi.org/10.1109/CEC.2018.8477921.
[91] F. Huang, A. Jolfaei, A.K. Bashir, Robust multimodal representation learning with evolutionary adversarial attention networks, IEEE Trans. Evol. Comput. 25 (5) (2021) 856–868, https://doi.org/10.1109/TEVC.2021.3066285.

[92] J.Y. Li, Z.H. Zhan, J. Xu, S. Kwong, J. Zhang, Surrogate-assisted hybrid-model estimation of distribution algorithm for mixed-variable hyperparameters optimization in convolutional neural networks, IEEE Trans. Neural Networks Learn. Syst. (2021), https://doi.org/10.1109/TNNLS.2021.3106399.
[93] M. Suganuma, S. Shirakawa, T. Nagao, A genetic programming approach to designing convolutional neural network architectures, in: Proc. Genet. Evol. Comput. Conf., 2017, pp. 497–504. https://doi.org/10.1145/3071178.3071229.
[94] B. Cao, J. Zhao, Z. Lv, Y.u. Gu, P. Yang, S.K. Halgamuge, Multiobjective evolution of fuzzy rough neural network via distributed parallelism for stock prediction, IEEE Trans. Fuzzy Syst. 28 (5) (2020) 939–952, https://doi.org/10.1109/TFUZZ.2020.2972207.
[95] Z. Lu, I. Whalen, Y. Dhebar, K. Deb, E.D. Goodman, W. Banzhaf, V.N. Boddeti, Multiobjective evolutionary design of deep convolutional neural networks for image classification, IEEE Trans. Evol. Comput. 25 (2) (2021) 277–291, https://doi.org/10.1109/TEVC.2020.3024708.
[96] Z. Lu, G. Sreekumar, E. Goodman, W. Banzhaf, K. Deb, V.N. Boddeti, Neural architecture transfer, IEEE Trans. Pattern Anal. Mach. Intell. 43 (9) (2021) 2971–2989, https://doi.org/10.1109/TPAMI.2021.3052758.
[97] J. Liu, M. Gong, Q. Miao, X. Wang, H. Li, Structure learning for deep neural networks based on multiobjective optimization, IEEE Trans. Neural Networks Learn. Syst. 29 (6) (2018) 2450–2463, https://doi.org/10.1109/TNNLS.2017.2695223.
[98] Z. Yang, Y. Wang, X. Chen, B. Shi, C. Xu, C. Xu, Q. Tian, C. Xu, CARS: Continuous evolution for efficient neural architecture search, in, Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (2020) 1829–1838.
[99] J. Liu, Y. Jin, Multi-objective search of robust neural architectures against multiple types of adversarial attacks, Neurocomputing 453 (2021) 73–84, https://doi.org/10.1016/j.neucom.2021.04.111.
[100] Y.-W. Wen, S.-H. Peng, C.-K. Ting, Two-stage evolutionary neural architecture search for transfer learning, IEEE Trans. Evol. Comput. 25 (5) (2021) 928–940, https://doi.org/10.1109/TEVC.2021.3097937.
[101] J.-Y. Li, Z.-H. Zhan, J. Zhang, Evolutionary computation for expensive optimization: A survey, Mach. Intell. Res. 19 (1) (2022) 3–23, https://doi.org/10.1007/s11633-022-1317-4.
[102] M. Gong, J. Liu, A.K. Qin, K. Zhao, K.C. Tan, Evolving deep neural networks via cooperative coevolution with backpropagation, IEEE Trans. Neural Networks Learn. Syst. 32 (1) (2021) 420–434.
[103] J. Long, S. Zhang, C. Li, Evolving deep Echo state networks for intelligent fault diagnosis, IEEE Trans. Ind. Informatics. 16 (7) (2020) 4928–4937, https://doi.org/10.1109/TII.2019.2938884.
[104] W.u. Deng, H. Liu, J. Xu, H. Zhao, Y. Song, An improved quantum-inspired differential evolution algorithm for deep belief network, IEEE Trans. Instrum. Meas. 69 (10) (2020) 7319–7327, https://doi.org/10.1109/TIM.2020.2983233.
[105] P. Cui, B. Shabash, K.C. Wiese, EvoDNN – An evolutionary deep neural network with heterogeneous activation functions, in: Proc. IEEE Congr. Evol. Comput., 2019, pp. 2362–2369. https://doi.org/10.1109/CEC.2019.8789964.
[106] S. Chen, W. Wang, B. Xia, X. You, Q. Peng, Z. Cao, W. Ding, CDE-GAN: Cooperative dual evolution-based generative adversarial network, IEEE Trans. Evol. Comput. 25 (5) (2021) 986–1000, https://doi.org/10.1109/TEVC.2021.3068842.
[107] Y. Sun, B. Xue, M. Zhang, G.G. Yen, Evolving deep convolutional neural networks for image classification, IEEE Trans. Evol. Comput. 24 (2) (2020) 394–407, https://doi.org/10.1109/TEVC.2019.2916183.
[108] A. Kenny, X. Li, A study on pre-training deep neural networks using particle swarm optimisation, Lect. Notes Comput. Sci. (2017) 361–372, https://doi.org/10.1007/978-3-319-68759-9_30.
[109] Y. Sun, H. Wang, B. Xue, Y. Jin, G.G. Yen, M. Zhang, Surrogate-assisted evolutionary deep learning using an end-to-end random forest-based performance predictor, IEEE Trans. Evol. Comput. 24 (2) (2020) 350–364, https://doi.org/10.1109/TEVC.2019.2924461.
[110] Y. Zhou, Y. Jin, J. Ding, Surrogate-assisted evolutionary search of spiking neural architectures in liquid state machines, Neurocomputing 406 (2020) 12–23, https://doi.org/10.1016/j.neucom.2020.04.079.
[111] M. Zhang, H. Li, S. Pan, J. Lyu, S. Ling, S. Su, Convolutional neural networks-based lung nodule classification: A surrogate-assisted evolutionary algorithm for hyperparameter optimization, IEEE Trans. Evol. Comput. 25 (5) (2021) 869–882, https://doi.org/10.1109/TEVC.2021.3060833.
[112] B. Wang, Y. Sun, B. Xue, M. Zhang, Evolving deep convolutional neural networks by variable-length particle swarm optimization for image classification, in: Proc. IEEE Congr. Evol. Comput., IEEE, 2018, pp. 1–8. https://doi.org/10.1109/CEC.2018.8477735.
[113] H. Zhu, Y. Jin, Multi-Objective Evolutionary federated learning, IEEE Trans. Neural Networks Learn. Syst. 31 (4) (2020) 1310–1322, https://doi.org/10.1109/TNNLS.2019.2919699.
[114] W. Irwin-Harris, Y. Sun, B. Xue, M. Zhang, A graph-based encoding for evolutionary convolutional neural network architecture design, Proc: IEEE Congr. Evol. Comput. CEC (2019) 546–553, https://doi.org/10.1109/CEC.2019.8790093.
[115] Z.-H. Zhan, X.-F. Liu, H. Zhang, Z. Yu, J. Weng, Y. Li, T. Gu, J. Zhang, Cloudde: A heterogeneous differential evolution algorithm and its distributed cloud version, IEEE Trans. Parallel Distrib. Syst. 28 (3) (2017) 704–716, https://doi.org/10.1109/TPDS.2016.2597826.
[116] J.-Y. Li, Z.-H. Zhan, R.-D. Liu, C. Wang, S. Kwong, J. Zhang, Generation-level parallelism for evolutionary computation: A pipeline-based parallel particle swarm optimization, IEEE Trans. Cybern. 51 (10) (2021) 4848–4859, https://doi.org/10.1109/TCYB.2020.3028070.
[117] Z.-J. Wang, Z.-H. Zhan, W.-J. Yu, Y. Lin, J. Zhang, T.-L. Gu, J. Zhang, Dynamic group learning distributed particle swarm optimization for large-scale optimization and its application in cloud workflow scheduling, IEEE Trans. Cybern. 50 (6) (2020) 2715–2729, https://doi.org/10.1109/TCYB.2019.2933499.
[118] Z.-H. Zhan, Z.-J. Wang, H.u. Jin, J. Zhang, Adaptive distributed differential evolution, IEEE Trans. Cybern. 50 (11) (2020) 4633–4647, https://doi.org/10.1109/TCYB.2019.2944873.
[119] Y. Guo, J.-Y. Li, Z.-H. Zhan, Efficient hyperparameter optimization for convolution neural networks in deep learning: A distributed particle swarm optimization approach, Cybern. Syst. 52 (1) (2021) 36–57, https://doi.org/10.1080/01969722.2020.1827797.
[120] T. Moriya, T. Tanaka, T. Shinozaki, S. Watanabe, K. Duh, Evolution-strategy-based automation of system development for high-performance speech recognition, IEEE/ACM Trans. Audio Speech Lang. Process. 27 (1) (2019) 77–88, https://doi.org/10.1109/TASLP.2018.2871755.
[121] S.C. Liu, Z.H. Zhan, K.C. Tan, J. Zhang, A multiobjective framework for many-objective optimization, IEEE Trans. Cybern. (2021) 1–15, https://doi.org/10.1109/TCYB.2021.3082200.
[122] X. Zhang, Z.H. Zhan, W. Fang, P. Qian, J. Zhang, Multi population ant colony system with knowledge based local searches for multiobjective supply chain configuration, IEEE Trans. Evol. Comput. (2021), https://doi.org/10.1109/TEVC.2021.3097339.
[123] H. Zhao, Z.-H. Zhan, Y. Lin, X. Chen, X.-N. Luo, J. Zhang, S. Kwong, J. Zhang, Local binary pattern-based adaptive differential evolution for multimodal optimization problems, IEEE Trans. Cybern. 50 (7) (2020) 3343–3357, https://doi.org/10.1109/TCYB.2019.2927780.
[124] Z.-G. Chen, Z.-H. Zhan, H. Wang, J. Zhang, Distributed individuals for multiple peaks: A novel differential evolution for multimodal optimization problems, IEEE Trans. Evol. Comput. 24 (4) (2020) 708–719, https://doi.org/10.1109/TEVC.2019.2944180.
[125] J.Y. Li, K.J. Du, Z.H. Zhan, H. Wang, J. Zhang, Multi-criteria differential evolution: Treating multitask optimization as multi-criteria optimization, Proc. Genet. Evol. Comput. Conf. (2021) 183–184, https://doi.org/10.1145/3449726.3459456.
[126] J. Zhang, C. Li, Adversarial examples: Opportunities and challenges, IEEE Trans. Neural Networks Learn. Syst. 31 (2020) 2578–2593, https://doi.org/10.1109/TNNLS.2019.2933524.
[127] J. Su, D.V. Vargas, K. Sakurai, One pixel attack for fooling deep neural networks, IEEE Trans. Evol. Comput. 23 (5) (2019) 828–841, https://doi.org/10.1109/TEVC.2019.2890858.
[128] P. Vidnerová, R. Neruda, Vulnerability of classifiers to evolutionary generated adversarial examples, Neural Networks 127 (2020) 168–181, https://doi.org/10.1016/j.neunet.2020.04.015.
[129] Y. Deng, C. Zhang, X. Wang, A multi-objective examples generation approach to fool the deep neural networks in the black-box scenario, in: Proc. IEEE Int. Conf. Data Sci. Cyberspace, 2019, pp. 92–99. https://doi.org/10.1109/DSC.2019.00022.
[130] Y.J. Gong, J. Zhang, Y. Zhou, Learning multimodal parameters: A bare-bones niching differential evolution approach, IEEE Trans. Neural Networks Learn. Syst. 29 (2018) 2944–2959, https://doi.org/10.1109/TNNLS.2017.2708712.
[131] C. Zhang, P. Lim, A.K. Qin, K.C. Tan, Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics, IEEE Trans. Neural Networks Learn. Syst. 28 (10) (2017) 2306–2318, https://doi.org/10.1109/TNNLS.2016.2582798.
[132] C. Zhang, J.H. Sun, K.C. Tan, Deep belief networks ensemble with multi-objective optimization for failure diagnosis, in: Proc. IEEE Int. Conf. Syst. Man, Cybern., 2016, pp. 32–37. https://doi.org/10.1109/SMC.2015.19.
[133] B. Fielding, T. Lawrence, L. Zhang, Evolving and ensembling deep CNN architectures for image classification, in: Proc. Int. Jt. Conf. Neural Networks, 2019, pp. 1–8. https://doi.org/10.1109/IJCNN.2019.8852369.
[134] Y. Zhou, G.G. Yen, Z. Yi, Evolutionary compression of deep neural networks for biomedical image segmentation, IEEE Trans. Neural Networks Learn. Syst. 31 (8) (2020) 2916–2929, https://doi.org/10.1109/TNNLS.2019.2933879.
[135] H. Shu, Y. Wang, X. Jia, K. Han, H. Chen, C. Xu, Q. Tian, C. Xu, Co-evolutionary compression for unpaired image translation, in: Proc. IEEE Int. Conf. Comput. Vis., 2019, pp. 3234–3243. https://doi.org/10.1109/ICCV.2019.00333.
[136] T. Wu, J. Shi, D. Zhou, Y. Lei, M. Gong, A multi-objective particle swarm optimization for neural networks pruning, in: IEEE Congr. Evol. Comput., 2019, pp.570–577. https://doi.org/10.1109/CEC.2019.8790145.
[137] D. Liang, Z.-H. Zhan, Y. Zhang, J. Zhang, An efficient ant colony system approach for new energy vehicle dispatch problem, IEEE Trans. Intell. Transp. Syst. 21 (11) (2020) 4784–4797, https://doi.org/10.1109/TITS.2019.2946711.
[138] X.-F. Liu, Z.-H. Zhan, J.D. Deng, Y. Li, T. Gu, J. Zhang, An energy efficient ant colony system for virtual machine placement in cloud computing, IEEE Trans. Evol. Comput. 22 (1) (2018) 113–128, https://doi.org/10.1109/TEVC.2016.2623803.
[139] S. Xie, A. Kirillov, R. Girshick, K. He, Exploring randomly wired neural networks for image recognition, in: Proc. IEEE Int. Conf. Comput. Vis., 2019, pp. 1284–1293. https://doi.org/10.1109/ICCV.2019.00137.
[140] Y. Duan, X. Chen, H. Xu, Z. Chen, X. Liang, T. Zhang, Z. Li, TransNAS-Bench-101: Improving transferability and generalizability of cross-task neural architecture search, Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (2021) 5251–5260.

[141] J.-R. Jian, Z.-H. Zhan, J. Zhang, Large-scale evolutionary optimization: a survey and experimental comparative study, Int. J. Mach. Learn. Cybern. 11 (3) (2020) 729–745, https://doi.org/10.1007/s13042-019-01030-4.
[142] X.-F. Liu, Z.-H. Zhan, T.-L. Gu, S. Kwong, Z. Lu, H.-L. Duh, J. Zhang, Neural network-based information transfer for dynamic optimization, IEEE Trans. Neural Networks Learn. Syst. 31 (5) (2020) 1557–1570, https://doi.org/10.1109/TNNLS.2019.2920887.
[143] Y. Jiang, Z.H. Zhan, K.C. Tan, J. Zhang, Optimizing niching centers for multimodal optimization, IEEE Trans. Evol. Comput. 22 (2018) 894–908, https://doi.org/10.1109/TCYB.2021.3125362.
[144] Z.-J. Wang, Z.-H. Zhan, Y. Lin, W.-J. Yu, H.-Q. Yuan, T.-L. Gu, S. Kwong, J. Zhang, Dual-strategy differential evolution with affinity propagation clustering for multimodal optimization problems, IEEE Trans. Evol. Comput. 22 (6) (2018) 894–908, https://doi.org/10.1109/TEVC.2017.2769108.
[145] T. Elsken, B. Staffler, J.H. Metzen, F. Hutter, Meta-learning of neural architectures for few-shot learning, in: Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recogn., 2020, pp. 12362–12372, https://doi.org/10.1109/CVPR42600.2020.01238.
[146] S. Xue, H. Chen, C. Xie, B. Zhang, X. Gong, D. Doermann, Fast and unsupervised neural architecture evolution for visual representation learning, IEEE Comput. Intell. Mag. 16 (3) (2021) 22–32.
[147] C. Wei, Y. Tang, C.N. Chuang Niu, H. Hu, Y. Wang, J. Liang, Self-supervised representation learning for evolutionary neural architecture search, IEEE Comput. Intell. Mag. 16 (3) (2021) 33–49.
[148] X.F. Liu, Z.H. Zhan, J. Zhang, Resource-aware distributed differential evolution for training expensive neural-network-based controller in power electronic circuit, IEEE Trans. Neural Networks Learn. Syst. (2021) 1–11, https://doi.org/10.1109/TNNLS.2021.3075205.
[149] Z. Tan, H. Wang, S. Liu, Multi-stage dimension reduction for expensive sparse multi-objective optimization problems, Neurocomputing 440 (2021) 159–174, https://doi.org/10.1016/j.neucom.2021.01.115.
[150] S.-H. Wu, Z.-H. Zhan, J. Zhang, SAFE: Scale-adaptive fitness evaluation method for expensive optimization problems, IEEE Trans. Evol. Comput. 25 (3) (2021) 478–491, https://doi.org/10.1109/TEVC.2021.3051608.
[151] Y. Jin, H. Wang, T. Chugh, D. Guo, K. Miettinen, Data-driven evolutionary optimization: An overview and case studies, IEEE Trans. Evol. Comput. 23 (3) (2019) 442–458, https://doi.org/10.1109/TEVC.2018.2869001.
[152] Y. Jin, H. Wang, C. Sun, Data-Driven Evolutionary Optimization, Springer, 2021.
[153] J.-Y. Li, Z.-H. Zhan, C. Wang, H.u. Jin, J. Zhang, Boosting data-driven evolutionary algorithm with localized data generation, IEEE Trans. Evol. Comput. 24 (5) (2020) 923–937, https://doi.org/10.1109/TEVC.2020.2979740.
[154] J.-Y. Li, Z.-H. Zhan, H. Wang, J. Zhang, Data-driven evolutionary algorithm with perturbation-based ensemble surrogates, IEEE Trans. Cybern. 51 (8) (2021) 3925–3937, https://doi.org/10.1109/TCYB.2020.3008280.
[155] Z. Zhang, L. Li, J. Lu, Gradient-based fly immune visual recurrent neural network solving large-scale global optimization, Neurocomputing 454 (2021) 238–253, https://doi.org/10.1016/j.neucom.2021.05.002.
[156] J.-R. Jian, Z.-G. Chen, Z.-H. Zhan, J. Zhang, Region encoding helps evolutionary computation evolve faster: A new solution encoding scheme in particle swarm for large-scale optimization, IEEE Trans. Evol. Comput. 25 (4) (2021) 779–793, https://doi.org/10.1109/TEVC.2021.3065659.
[157] Y. Han, X. Gu, Cooperative hybrid evolutionary algorithm for large scale multi-stage multi-product batch plants scheduling problem, Neurocomputing 419 (2021) 80–96, https://doi.org/10.1016/j.neucom.2020.07.094.
[158] Y.-F. Ge, W.-J. Yu, Y. Lin, Y.-J. Gong, Z.-H. Zhan, W.-N. Chen, J. Zhang, Distributed differential evolution based on adaptive mergence and split for large-scale optimization, IEEE Trans. Cybern. 48 (7) (2018) 2166–2180, https://doi.org/10.1109/TCYB.2017.2728725.
[159] Z.-J. Wang, Z.-H. Zhan, S. Kwong, H.u. Jin, J. Zhang, Adaptive granularity learning distributed particle swarm optimization for large-scale optimization, IEEE Trans. Cybern. 51 (3) (2021) 1175–1188, https://doi.org/10.1109/TCYB.2020.2977956.
[160] B. Evans, H. Al-Sahaf, B. Xue, M. Zhang, Evolutionary deep learning: A genetic programming approach to image classification, in: Proc. IEEE Congr. Evol. Comput. CEC, 2018, pp. 1-8. https://doi.org/10.1109/CEC.2018.8477933.

Zhi-Hui Zhan (Senior Member, IEEE) received the Bachelor's degree and the Ph. D. degree in Computer Science from the Sun Yat-Sen University, Guangzhou China, in 2007 and 2013, respectively.
He is currently the Changjiang Scholar Young Professor with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. His current research interests include evolutionary computation, swarm intelligence, machine learning including deep learning, and their applications in real-world problems and in environments of cloud computing and big data.
Dr. Zhan was a recipient of the IEEE Computational Intelligence Society (CIS) Outstanding Early Career Award in 2021, the Outstanding Youth Science Foundation from National Natural Science Foundations of China (NSFC) in 2018, and the Wu Wen-Jun Artificial Intelligence Excellent Youth from the Chinese Association for Artificial Intelligence in 2017. His doctoral dissertation was awarded the IEEE CIS Outstanding Ph. D. Dissertation and the China Computer Federation Outstanding Ph. D. Dissertation. He is listed as one of the World's Top 2% Scientists for both Career-Long Impact and Year Impact in Artificial Intelligence and one of the Highly Cited Chinese Researchers in Computer Science. He is currently an Associate Editor of the *IEEE Transactions on Evolutionary Computation*, the *Neurocomputing*, and the *Memetic Computing*.

Jian-Yu Li (Student Member, IEEE) received the B.S. degree in computer science and technology from South China University of Technology, Guangzhou, China, in 2018, where he is currently pursuing the Ph.D. degree in computer science and technology with the School of Computer Science and Engineering.
His research interests mainly include computational intelligence, data-driven optimization, machine learning including deep learning, and their applications in real-world problems, and in environments of distributed computing and big data. He has been invited as a reviewer of the *IEEE Transactions on Evolutionary Computation* and the *Neurocomputing*.

Jun Zhang (Fellow, IEEE) received the Ph.D. degree from the City University of Hong Kong in 2002.
He is currently a Korea Brain Pool Fellow Professor with Hanyang University, South Korea, and a visiting Professor with Chaoyang University of Technology. His current research interests include computational intelligence, cloud computing, operations research, and power electronic circuits. He has published over more than 150 IEEE Transactions papers in his research areas.
Dr. Zhang was a recipient of the Changjiang Chair Professor from the Ministry of Education, China, in 2013, The National Science Fund for Distinguished Young Scholars of China in 2011 and the First-Grade Award in Natural Science Research from the Ministry of Education, China, in 2009. He is currently an Associate Editor of the *IEEE Transactions on Evolutionary Computation* and the *IEEE Transactions on Cybernetics*.