Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information website.

Applied Soft Computing 111 (2021) 107675

Contents lists available at ScienceDirect

Applied Soft Computing

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

An oppositional-Cauchy based GSK evolutionary algorithm with a novel deep ensemble reinforcement learning strategy for COVID-19 diagnosis

Seyed Mohammad Jafar Jalali [a], Milad Ahmadian [b], Sajad Ahmadian [c,*], Abbas Khosravi [a], Mamoun Alazab [d], Saeid Nahavandi [a]

[a] Institute for Intelligent Systems Research and Innovation, (IISRI), Deakin University, Geelong, Australia
[b] Deparment of Computer Engineering, Razi University, Kermanshah, Iran
[c] Faculty of Information Technology, Kermanshah University of Technology, Kermanshah, Iran
[d] College of Engineering, IT and Environment, Charles Darwin University, Australia

ARTICLE INFO

Article history:
Received 27 December 2020
Received in revised form 8 May 2021
Accepted 28 June 2021
Available online 10 July 2021

Keywords:
COVID-19
Deep convolutional neural network
Evolutionary computation
Optimization
Deep reinforcement learning
Image classification

ABSTRACT

A novel coronavirus (COVID-19) has globally attracted attention as a severe respiratory condition. The epidemic has been first tracked in Wuhan, China, and has progressively been expanded in the entire world. The growing expansion of COVID-19 around the globe has made X-ray images crucial for accelerated diagnostics. Therefore, an effective computerized system must be established as a matter of urgency, to facilitate health care professionals in recognizing X-ray images from COVID-19 patients. In this work, we design a novel artificial intelligent-based automated X-ray image analysis framework based on an ensemble of deep optimized convolutional neural networks (CNNs) in order to distinguish coronavirus patients from non-patients. By developing a modified version of gaining–sharing knowledge (GSK) optimization algorithm using the Opposition-based learning (OBL) and Cauchy mutation operators, the architectures of the deployed deep CNNs are optimized automatically without performing the general trial and error procedures. After obtaining the optimized CNNs, it is also very critical to identify how to decrease the number of ensemble deep CNN classifiers to ensure the classification effectiveness. To this end, a selective ensemble approach is proposed for COVID-19 X-ray based image classification using a deep Q network that combines reinforcement learning (RL) with the optimized CNNs. This approach increases the model performance in particular and therefore decreases the ensemble size of classifiers. The experimental results show that the proposed deep RL optimized ensemble approach has an excellent performance over two popular X-ray image based COVID-19 datasets. Our proposed advanced algorithm can accurately identify the COVID-19 patients from the normal individuals with a significant accuracy of 0.991441, precision of 0.993568, recall (sensitivity) of 0.981445, F-measure of 0.989666 and AUC of 0.990337 for Kaggle dataset as well as an excellent accuracy of 0.987742, precision of 0.984334, recall (sensitivity) of 0.989123, F-measure of 0.984939 and AUC of 0.988466 for Mendely dataset.

1. Introduction

In December 2019, in Wuhan, Hubei province, China, a number of cases regarding the pneumonia of unknown aetiology has been appeared. The virus origin of the pneumonia was seen to be triggered by "severe acute coronavirus syndrome 2" (SARS-CoV-2), with the affiliate condition later identified by the World Health Organisation as the 2019 coronavirus condition (COVID-19) [1]. The recently discovered virus is widely infectious and transmissible to the SARS-CoV and other widespread breathing viruses.

With regard to COVID-19 pandemic, the World Health Organisation has declared a global Emergency for public health outbreak of the novel Corona-virus infection. A global pandemic downward trend happened on early of March 2020 as the epidemic quickly progressed across the globe [2]. Those certain viral respiratory

* Corresponding author.
E-mail addresses: mohammadjj.it@gmail.com (S.M.J. Jalali), m.ahmadian@stu.razi.ac.ir (M. Ahmadian), s.ahmadian239@gmail.com (S. Ahmadian), abbas.khosravi@deakin.edu.au (A. Khosravi), m.alazab@icsl.com.au (M. Alazab), saeid.nahavandi@deakin.edu.au (S. Nahavandi).

https://doi.org/10.1016/j.asoc.2021.107675
1568-4946/

tract disorders such as in bacterial pneumonia and respiratory-syncytial virus are closely related in psychiatric symptoms of infected individuals with COVID-19 [3]. The most typical symptoms of patient individuals involve dry husk, chest tightness, fever, hypotension, vomiting, chronic fatigue, and stuffy nose which can be seen on clinical chest X-ray or CT images. Currently, no antiviral medicine or vaccine clinically authorized for COVID-19 is readily accessible [4,5]. Significant new occurrences of COVID-19 in around 200 countries with more intensity cases in South Korea, Japan, Italy, Germany, and France were reported in May 2020 [6,7]. A close interaction of individuals to individuals with droplets in the air is the most common source of new cases [8,9]. Highly infectious substances may also contribute to infection of nose, eyes, or mouth after touching. It is noticeable that an earlier COVID-19 diagnosis will disrupt the COVID-19 expansion and avoid dissemination of close interactions by early isolation of patients, monitoring, and locking-down [10,11]. Efficient control for the advancement of the disease is a crucial component in risk management of individuals infected by COVID-19.

Clinical imaging methodologies including the chest X-ray and computed tomography (CT) carry out a vital role in both the identification of positive COVID-19 pneumonia and in the observing of virus evolution. Such types of images play a critical role in diagnosis of the COVID-19 side effects. Furthermore, advanced machine learning techniques have shown their great significance progress in addressing many practical applications such as recovering low-rank tensors [12], feature selection [13–15], information retrieval [16–20], and computer vision [21,22,22,23,23,24,24,25]. By utilizing the excellent benefits of an advanced machine learning strategy called deep learning with analysis of X-ray and CT images can benefit health care providers with treatment assists for COVID-19, which leads in contributing to higher insights into the virus progresses. Convolutional neural networks (CNNs) as the powerful algorithms from deep learning family, thanks to their great advances in computing resources and accessible broad image databases in many real-world problems [25–29], have recently shown to surpass the performance of the medical practitioners for several medical applications [30,31].

In order to diagnose COVID-19 from X-ray images, Hemdan et al. [32] have designed a deep learning platform called COVIDX-Net. This platform is evaluated over 50 Chest X-ray images with 25 confirmed positive COVID-19 cases in which the authors obtained the F1-scores of 0.89 and 0.91 for normal and COVID-19 samples. Wang and Wong [33] developed a deep learning algorithm for COVID-19 named as COVID-Net in categories of the normal, pneumonium-bacterial and pneumonic of COVID-19. 83.5% accuracy of the classification model was acquired by their proposed COVID-Net. In another study conducted by Leoy et al. [34], researchers used the deep learning-based generative adversarial network (GAN) algorithm to significantly boost training images of four classes including normal, COVID-19, infectious pneumonia, and pathogenic pneumonia. Compared to other well-known deep learning architectures, the accuracy and sensitivity of their proposed model out performed the mentioned models. Most recently, Apostolopoulus and Mpesiana [35] have trained multiple deep learning algorithms in which they obtained the accuracy performance of 98.75% and 93.48% for two-class and three-class COVID-19 datasets, respectively. In another work, Narin et al. [36] have equipped the deep ResNet50 architecture with chest X-ray images and obtained the 98% accuracy classification performance for a two-classes COVID-19 dataset. Throughout the COVID-19 classification, Sethy and Behera [37] utilized a number of CNN models hybridized with the support vector machine (SVM) algorithm. Their research reveals that the hybrid SVM-ResNet50 algorithm outperforms by far the most effective performance among the compared deep learning models. In another work, a deep learning architecture based on DarkNet strategy, consisting of 17 convolutional layers with Leaky ReLU as the active function, was presented by Ozturk et al. [38]. Their framework obtained accuracy for binary classes with 98.08% and multi-class cases with 87.02%, respectively. In the work presented by [39], the authors utilized the features of three deep learning algorithms in order to detect COVID-19 patients using lung X-ray images in which their proposed architecture outperforms well by showing higher accuracy and sensitivity evaluation metrics. Goel et al. in [40] proposed a model for automatic diagnosis of COVID-19 images called as OptCoNet. In this framework, the proposed model composed of optimized feature extraction and classification components. They also used the basic version of evolutionary gray wolf optimizer (GWO) for optimizing only four hyperparameters of the CNN layers. This research also outperforms well using several various evaluation metrics for COVID-19 diagnosis.

Although previous studies have shown considerable accuracy in recognizing images related to COVID-19, in all of these studies, the design of deep learning networks except the work of [40] has been done manually, which is a very time consuming and tedious task. It should be mentioned that even in [40], the authors just optimized four CNN hyperparameters including learning rate, number of epochs, momentum, and regularization coefficient in which the other critical hyperparameters for CNN designing that have huge impact on the performance of CNNs such as kernel size, number of filters and batch size were considered as fixed values. Thus, the fully-automatic design of the architecture of deep learning networks in an optimal way, which is done in the shortest time with the least trial and error process [41,42], is an essential task in designing the architecture of deep neural networks [43,44].

The evolutionary computational algorithms have become more widely known in recent decades thanks to their high ability for adaption and flexibility toward tackling complex real-world problems [45–48]. Several traditional and modern evolutionary algorithms in the recent years are proposed in which their famous ones are regarded as ant colony [49], as well as gray wolf optimizer [50], fruit fly optimization algorithm [51], moth–flame optimizer [52] and Harris hawks optimization [53] which have been effectively used in a wide range of applications [54]. Deep neuroevolution is an efficient concept that is based on designing the architecture of deep neural networks optimally and automatically based on the robust evolutionary computational algorithms [27,55,56]. On the other hand, ensemble techniques can increase the performance capability of the classifier by integrating multiple classification algorithms. Recent years have received widespread attention from academics by the deep reinforcement learning model, an artificial intelligence approach with a strong optimizing capability. Thanks to the high performance of DNE in the domain of image classification, the compatibility of ensemble learning models and the robust capacity to learn after integrating DNE with deep reinforcement Learning, an ensemble deep reinforcement learning framework based on deep neuroevolution classifiers is investigated in this research study.

To be specific, in this study, to estimate CNN hyperparameters accurately and efficiently, we design an improved Gaining–Sharing Knowledge optimization algorithm (IGSK) which is on account of the opposition-based learning strategy and the Cauchy mutation mechanism. Our proposed IGSK algorithm has excellent performance, finds global optima efficiently, mature convergence, and makes an effective balance between exploration and exploitation phases. Then, the deep reinforcement learning approach is employed to obtain an optimal subset of classifiers which can be used as the final selected base classifiers in the ensemble strategy. Finally, by examining the performance of our proposed robust approach on two well-known image classification COVID-19 datasets, it noticeably outperforms a number of

well-known algorithms as a feasible hyperparameter classification model for COVID-19 diagnosis and demonstrates promising functionalities. In order to distinguish the significant difference between our proposed algorithm with the related reviewed works for COVID-19 diagnosis, a taxonomy table is presented in Table 1. According to this table, we observe that among the related works carried out to diagnose COVID-19 using the deep learning technologies, our proposed framework is the only model in the relevant literature that uses the deep reinforcement learning strategy to ensemble the sets of optimized deep CNN models using an improved evolutionary algorithm for COVID-19 diagnosis.

In summary, the contribution and innovation of this work are highlighted in details as follows:

1. We design an advanced evolutionary algorithm which has been modified by two strong optimization operators including opposition-based learning (OBL) and Cauchy mutation operators over the original version of Gaining–Sharing knowledge optimization algorithm (GSK) to converge faster and increase more the exploration and exploitation phases during the optimization procedure for finding the optimal solutions. Our proposed model tunes the eleven convolutional neural network hyperparameters efficiently which helps in increasing the performance of CNN architectures and automatically designs the network architecture without the need of manually designing.

2. For the first time, the deep reinforcement learning algorithm is used to combine several deep neuroevolution models for the purpose of COVID-19 diagnosis. Unlike the conventional ensemble learning models, the reinforcement learning approach is a technique that has a consistent potential to be intelligent via the communication between the agent and the environment, allowing this method more competitive in the compatibility and optimization process. In addition, the incorporation of several deep neuroevolution models can effectively increase the flexibility and scalability of the proposed framework.

3. We analyze the performance of our proposed deep ensemble model over two well-known X-ray based image classification datasets using eight powerful evolutionary as well as deep state of the art algorithms. The results of this extensive analysis show that the proposed deep neuroevolution-based RL model has the best performance among the compared deep learning benchmarks for COVID-19 diagnosis.

The rest of the paper is as follows: in Section 2, the proposed evolutionary-based deep RL methodology is explained in details. The experimental setups, two utilized datasets and the experimental results are described in Section 3. The strengths and weaknesses of the proposed method are discussed in Section 4. We finally conclude this work in Section 5.

2. Methodology

In this section, the proposed ensemble image classification approach is introduced. In this method, we use convolutional neural network (CNN) as the base classifier to detect COVID-19 cases. To obtain the optimal values of the CNN's hyperparameters, a new evolutionary algorithm is proposed based on the Gaining–Sharing Knowledge (GSK) optimization algorithm. Moreover, the reinforcement learning model is used to make an ensemble classifier method by selecting an optimal subset of the original base classifiers set. In the following subsections, we first give an overview of the CNN model. Then, the proposed evolutionary algorithm is discussed. Finally, the proposed ensemble image classification method is introduced in details.

2.1. Convolutional neural network

Convolutional Neural Network is nowadays recognized as one of the most prominently applicable machine learning technologies, in particular in applications relevant to machine vision. CNNs are able to learn representations from grid of pixels and have recently seen major improvements in performance for several real-world applications. As CNNs acquire both successful feature extraction and discrimination capability, they are used for the feature extraction and classification tasks in the standard machine learning frameworks. A standard CNN architecture mainly involves multiple convolution and pooling layers, and at the end, one or more fully-connected layers. In some cases, simply a fully-connected layer has been substituted by a global max or average pooling layer. Apart from certain mapping functions, various regulatory units such as batch norms and dropouts are also utilized to improve CNN efficiency. In developing new architectures and thus achieving improved performance, the laying-out of CNN components plays a vital factor. The mechanism of these components in CNN architecture is discussed in the following subsections.

2.1.1. Convolutional layer

The convolutional layer consists of a set of convolutional kernels in which each neuron represents a kernel. Convolutional kernel operates by splitting the image into small pieces, typically referred to as receptive fields. The division of an image into small pieces allows feature patterns to be extracted. The following equation can describe the convolution operation process:

$$f_l^k(p, q) = \sum_c \sum_{x,y} i_c(x, y).e_l^k(u, v) \tag{1}$$

where $f_l^k(p, q)$ represents component (p, q) of feature matrix in which $i_c(x, y)$ denotes to the cth channel of the tensor input image that is wisely multiplied by $e_l^k(u, v)$ which is component (u, v) of the kth convolutional kernel for the lth layer. p and q are pth and qth row and column of feature matrix respectively, whereas x and y are xth and yth coordinators under consideration of an image. Finally, the output feature-map (F_l^k) of the input feature matrix of the lth layer and kth neuron can be computed as follows:

$$\mathbf{F}_l^k = \left[f_l^k(1, 1), \ldots, f_l^k(p, q), \ldots, f_l^k(P, Q)\right] \tag{2}$$

where P and Q represent total number of rows and columns of feature matrix, respectively.

Because of the weight sharing capabilities of the convolutional operation, the sliding kernel with the same weight on the image can extract different sets of features in an image, thus making CNN more powerful than fully connected networks. Convolution operation can also be classified into various types depending on the size and type of filters, padding type, and convolution direction.

2.1.2. Pooling layer

Feature patterns obtained from the output of the convolution operation can appear at different areas in the image. Once features have been extracted, their exact locations would be less important as long as it remains approximate to others. Pooling or down sampling is an important local operation. This operation gathers similar information in the receptive area neighborhood and provides the dominant response within this area. The pooling operation (Z_l^k) is described as follows:

$$\mathbf{Z}_l^k = g_p(\mathbf{F}_l^k) \tag{3}$$

where F_l^k denotes to the pooled feature-map of lth layer for kth input feature-map. Moreover, $g_p(.)$ represents the type of pooling

Table 1
A taxonomy of the reviewed works for COVID-19 diagnosis compared to our proposed model.

Work	Optimization algorithm	Manual architecture design	Ensemble learning	Tuned hyperparameters										
				Kernel size	Number of filters	Number of epochs	Batch size	Number of convolutional layers	Dropoutrate	Maxpooling size	Learning rate	Momentum rate	Optimizer	Activation function
Hemdan et al. [32]	No	Yes	No	No	No	No	No	No	No	No	No	No	No	No
Wang and Wong [33]	No	Yes	No	No	No	No	No	No	No	No	No	No	No	No
Leoy et al. [34]	No	Yes	No	No	No	No	No	No	No	No	No	No	No	No
Apostolopoulus and Mpesiana [35]	No	Yes	No	No	No	No	No	No	No	No	No	No	No	No
Narin et al. [36]	No	Yes	No	No	No	No	No	No	No	No	No	No	No	No
Sethy and Behera [37]	No	Yes	No	No	No	No	No	No	No	No	No	No	No	No
Ozturk et al. [38]	No	Yes	No	No	No	No	No	No	No	No	No	No	No	No
Hassantabar et al. [39]	No	Yes	No	No	No	No	No	No	No	No	No	No	No	No
Goel et al. in [40]	Yes	No	No	No	No	Yes	No	No	No	No	Yes	Yes	No	No
Our work	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

operation. When using the pooling operation, a set of features can be extracted which are deterministic to translation changes and minor distortions. Reducing feature-map size to invariant feature sets not only controls network complexity, but also helps to improve overfitting generalization. CNN uses different types of poolings, such as average, max, overlapping, L2, etc.

2.1.3. Activation function

The activation function acts as a decision-making mechanism and helps to learn dynamic models. The selection of a suitable activation function can speed up the training procedure. Eq. (4) determines the activation function for a convolutional function map:

$$\mathbf{T}_l^k = g_a(\mathbf{F}_l^k) \tag{4}$$

where F_l^k is a convolution output assigned to the g_a (activation function), which adds non linearity to a T_l^k (output transformed) for the lth layer. Particular functions such as sigmoid, tanh, Maxout, SWISH, ReLU, and ReLU variants are used in the literature as the inculcation of non-linear variations of features. Still ReLU and its variants are favored because they help solving the problem of the vanishing gradient.

2.1.4. Batch normalization

Batch normalization is used in feature maps for addressing the issues with regards to the change of internal covariance. The shift in internal covariance is a change in the allocation of values of hidden units that decreases convergence (forcing a learning rate at a low value) and allows the parameters to be carefully initialized. Batch normalization is shown in Eq. (5) for a transformed function map F_l^k:

$$\mathbf{N}_l^k = \frac{\mathbf{F}_l^k - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}} \tag{5}$$

where $\mathbf{N}_l^k$ depicts the normalized feature-map, $\mathbf{F}_l^k$ represents the input feature-map, μ_B and σ_B^2 are the average and variance of a feature-map of a mini batch, respectively. ε is added for numerical stability to avoid divisions by zero. Through setting the feature-map values to a zero mean and unit variance, batch normalization enriches their distribution. This also smoothes the gradient flow and operates as a control factor which further helps to boost the network generalization.

2.1.5. Dropout

Dropout incorporates network regulations which effectively enhance the generalization of certain units or connections with a certain probability. Neural networks often co-adapt multiple connections which learn about a nonlinear connection and trigger overfitting. The random dropping of a number of units or connections generates a number of small network architectures and eventually a representative network with small weights is selected. This identified architecture is considered to approximate all the networks.

2.1.6. Fully connected layer

Fully-connected layer is typically used for classification at the end of the network. It is considered as a global operation, unlike pooling and convolution. The role of this operation is in collecting information from feature extraction phases and analyzing the performance of all the previous layers, globally. As a result, it generates a non-linear combination of the selected features used for data classification.

2.2. Proposed evolutionary algorithm: Improved GSK

Gaining–Sharing Knowledge (GSK) optimization algorithm is focused on the concept of knowledge sharing and gathering in humans' lives first proposed by [57]. This algorithm consists of two essential stages: junior gaining and sharing stage and senior gaining and sharing stage which are described in the following. Nearly every individual (people) within a specific population may communicate and constantly influence each other through collaboration and competition to be highly competent and skilled enough to tackle real-life situations and overcome complicated challenges. Yet the people must acquire information and share it with others if their desire is to be the experienced persons. Therefore, they can rapidly communicate their expertise and experience in various domains and with the best people who have successful personalities and behaviors. Below is the mathematical definition of the aforementioned principle of the gaining and sharing knowledge procedure. Assume $x_i, i = 1, 2, 3, \ldots, N$ represents the individuals of a given population, where N individuals are in this population and x_i denotes to $x_{i1}, x_{i2}, \ldots, x_{iD}$, i.e., D is the number of specialization fields and $f_i, i = 1, 2, 3, \ldots, N$ indicated their subsequent fitness values. At the start of the searching procedure, the number of dimensions (D) is calculated with the following non-linear formulation called experience equation, based on the basic principle of gaining–sharing knowledge.

$$D\,(juniorphase) = (problemsize) \times \left(1 - \frac{G}{GEN}\right)^k, \tag{6}$$

where k is a real positive number denotes to the knowledge rate. G represents the generation number and GEN denotes to the maximum number of generations.

$$D\,(seniorphase) = problemsize - D\,(juniorphase)\,. \tag{7}$$

As a result, in the initialization stage, the number of dimensions gained and shared by each vector using both schemes is determined. As suggested by the original work, we consider the problem size to 100 and k equal to 2.

In junior gaining–sharing knowledge stage, every individual is arranged according to their objective function value in the ascending order as $x_{best}, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_{worst}$. Then, two separate individuals (the closest individuals), the better (x_{i-1}), and worse (x_{i+1}) than the current individual (x_i) are chosen for each individual to form a gain source of knowledge. In addition, the algorithm selects a random source of knowledge sharing for another individual.

In senior gaining–sharing knowledge stage, after all individuals are sorted in an ascending order, they are divided into three categories including best, better (middle), and worst individuals according to their objective function. In addition, two randomly selected vectors of the top and bottom in the present population (NP) are used in each of the individuals in the proposed senior scheme, while one of the third vectors is randomly selected from the middle individuals to make up the sharing portion.

Opposition-based learning (OBL) has demonstrated encouraging improvements in several optimization algorithms as a growing interest metaheuristic strategy [58–60]. The advantage of this strategy is in taking into consideration both of the solution and its opposite in such a depth analysis of the search space resulting in improving the convergence rate of the metaheuristic algorithm [61]. Furthermore, the probability is strengthened to find the solution closely related to the global solution. Two definitions are required to describe the main components of OBL which are 1D opposite number and D-dimensional opposite point.

-Opposite number:
Assume h denotes to a real number in the $h \in [\text{lb}, \text{ub}]$ interval. The opposite number of h is defined by $\hat{h}$ as follows:

$$\hat{h} = \text{lb} + \text{ub} - h. \tag{8}$$

-Opposite point:
Assume $h = (h_1, h_2, \ldots, h_D)$ is a point in a D-dimensional space, in which $(h_1, h_2, \ldots, h_D) \in \mathbb{R}$, $h_i \in [\text{lb}_i, \text{ub}_i]$ and $\forall i \in (1, 2, \ldots, D)$. Therefore, the opposite point $\hat{h} = (\hat{h}_1, \hat{h}_2, \ldots \hat{h}_D)$ is given by:

$$\hat{h}_i = \text{lb}_i + \text{ub}_i - h_i, \quad i \in 1, 2, \ldots, D. \tag{9}$$

Based on these two concepts, we define the optimization procedure using the opposite population. Assume $f(h)$ indicates the fitness of the search agent $h = (h_1, h_2, \ldots, h_D)$ in a D-dimensional space where $f(\hat{h})$ represents its opposite fitness $\hat{h} = (\hat{h}_1, \hat{h}_2, \ldots \hat{h}_D)$. If $f(h) > f(\hat{h})$, then swap h with $\hat{h}$. Otherwise, Let the h unmodified. While the current and opposite points are considered for evaluation and calculation simultaneously, faster convergence into a better solution is observed.

In GSK algorithm, a procedure to control the performance of search agents is essential to help the search agents escape from local search locations by letting them move to better solutions. A feasible solution is on introducing Cauchy mutation which has been used in improving other metaheuristics such as differential evolution [62], particle swarm optimization [63], and moth–flame optimizer [64]. The Cauchy function can be formulated in a 1D space as follows:

$$f(x) = \frac{1}{\pi} \frac{a}{x^2 + a^2}, \quad a > 0 \ \& \ -\infty < x < \infty \tag{10}$$

where a variable denotes to the scale factor. The probability density for Cauchy function is defined as follows:

$$F(x) = \frac{1}{2} + \frac{1}{\pi} \arctan\left(\frac{x}{a}\right) \tag{11}$$

This mutation operator disrupts the population of search agents and motivates them to avoid into local optima. Cauchy mutation is combined with the previous improvement OBL in GSK as follows:

$$W_i = \frac{\left(\sum_{i=1}^{N} x_{ij}\right)}{N} \tag{12}$$

where W_i represents vector of the weights, N is the population size, and x_{ij} defines the jth position of the ith search agent.

$$x_j' = x_j + W_j \cdot R \tag{13}$$

where R represents a Cauchy distributed random number.

The flowchart of the improved GSK (IGSK) which is based on OBL and Cauchy mutation is shown in Fig. 1.

2.3. Deep reinforcement learning and evolved CNN based ensemble classifier

In this section, we aim to propose our novel deep neuroevolution (DNE) reinforcement learning (RL) ensemble image classifier model called as DNE-RL. The proposed method is based on three main phases. In the first phase, the Bagging method is utilized to generate a set of base CNN classifiers to employ in ensemble strategy. Then, in the second phase, the hyperparameters of the base CNN classifiers are optimized using the improved GSK algorithm. Finally, in the third phase, the reinforcement learning approach is employed to obtain an optimal subset of classifiers which can be used as the final selected base classifiers in the ensemble strategy. The overview of the proposed COVID-19 diagnosis mechanism is demonstrated in Fig. 2. The details of these three phases of the proposed method are discussed in the following subsections.

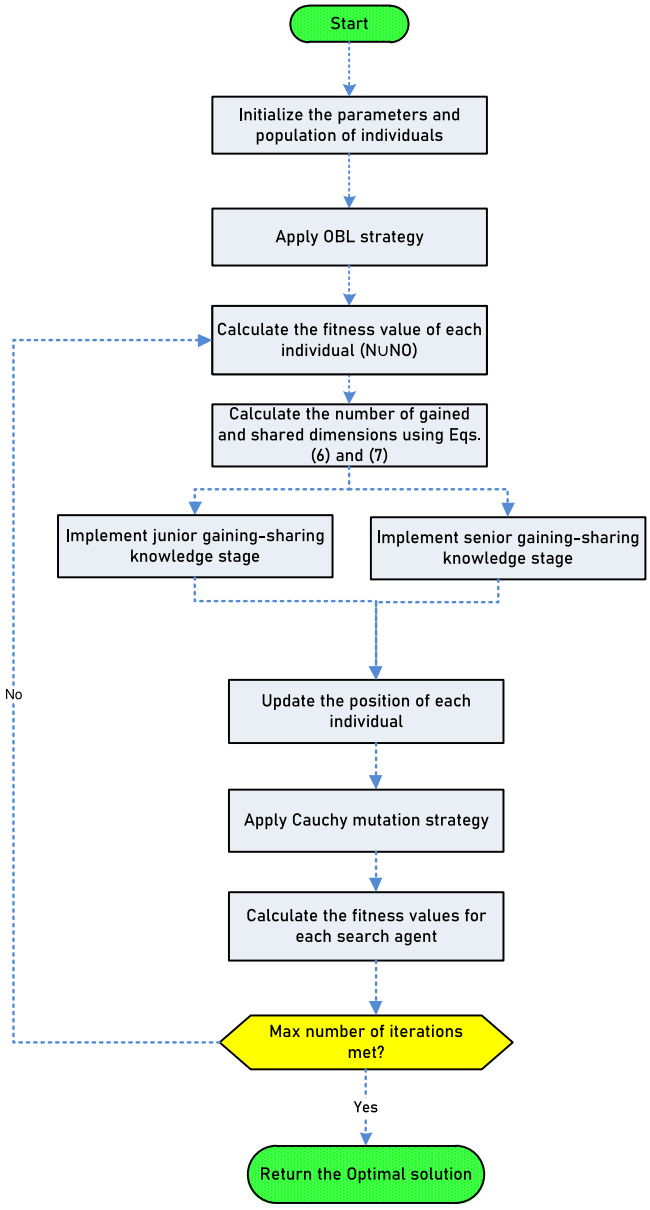


Fig. 1. The flowchart of the proposed IGSK algorithm.

2.3.1. Generation of base classifiers

In the proposed method, we use a homogeneous model to make an ensemble classifier approach. To this end, the Bagging method is used to generate a set of base classifiers. The motivation of choosing the Bagging method is the effectiveness and simplicity of its general process in generating the base classifiers. This method attempts to generate a set of base classifiers $B_L = \{b_1, b_2, \ldots, b_L\}$ based on selecting L training sets according to the original training set of the input samples. In other words, the Bagging method forms a training set for each base classifier by randomly selecting samples from the original training set. It is worth noting that the input samples in the proposed method are COVID/NON-COVID images which are used as the inputs of the base CNN classifiers. After generating the base CNN classifiers, each of them can be trained individually according to its training set.

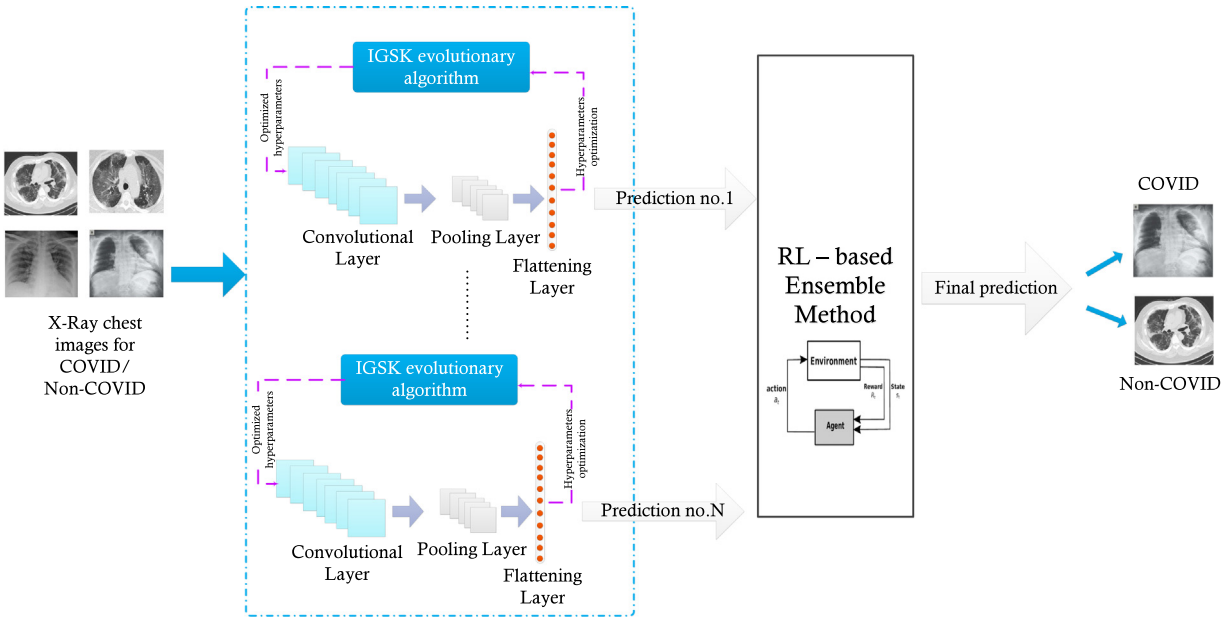


Fig. 2. The proposed DNE-RL framework for COVID-19 diagnosis.

2.3.2. Optimization of base classifiers

Generally, the performance of the CNN models depends on choosing the values of their hyperparameters. In other words, if the values of CNN hyperparameters are not appropriate, then its classification quality can be significantly declined. To address this problem, in this phase, the improved version of GSK algorithm as an evolutionary approach is utilized to obtain the optimal values of hyperparameters of the base CNN classifiers.

Two essential considerations, including solutions representation and fitness evaluation, should be properly considered for every task of evolutionary optimization. We cover eleven key hyperparameters in the proposed method: kernel size, number of filters, number of epochs, batch size, maxpooling size, dropout rate, learning rate, momentum rate, and number of convolutional layers which need to be optimized by the IGSK strategy. In IGSK, all solutions can therefore be represented as an eleven-dimensional vector, each referring to one of the eleven CNN hyperparameters. Dropout rate, momentum rate, and learning rate can be accomplished through using IGSK to obtain the hyperparameters with continuous values. In addition, other hyperparameters with discrete attributes include kernel size, number of convolutional layers, number of epochs, number of filters, batch size, and maxpooling size. As IGSK repeatedly explores the space for solutions, we require to map the achieved values optimally into their relative discrete attributes for these hyperparameters. This motivates us to design an effective methodology to map any real values into their corresponding integer values. To this reason, the continuous values for every hyperparameter are passed as discrete search space to $D = [K_1, K_2, \ldots, K_n]$. For formulating the discretization model , the following equations are utilized:

$$\alpha = 1 + n \times R \tag{14}$$

$$\beta = \min(\lfloor \alpha \rfloor, n) \tag{15}$$

where R represents a real attribute within the $[0, 1]$ interval to be traversed through continuous search space. The α is a mapping between R and $[1, n + 1]$, and the mapping from α to $[1, 1, 2, \ldots, n]$ is used by β. Therefore, the integer values of all solutions can be determined by the following equation, referring to a continuous dimension:

$$X_{ij} = K_\beta \tag{16}$$

A random initialization of n solutions in population with Eq. (6) is provided by the IGSK-CNN algorithm. The eleven-dimensional $X_{ij}, i = 1, \ldots, n$ and $j = 1, \ldots, 11$ is marked on each solution, where j for each dimension has one of eleven CNN hyperparameters.

The continuous updating of current solutions with Eq. (8) can provide new solutions after the initial population has been generated. At GSK's initialization stage, we use OBL technique to significantly boost its search space population diversity through Eq. (9). Beside, we incorporate the Cauchy mutation technique into GSK for increasing its convergence speed to generate solutions in a faster manner.

The whole procedure is continued once the final condition is fulfilled and the best solution is justified as the intended result. The optimum CNN hyperparameters can be used for this solution. To examine the performance of all the solutions, we need to have a fitness function. The data on the image dataset COVID-19 is segmented into two distinct training and test sets to accomplish this goal. The training set is originally intended to optimize the hyperparameters of the CNNs using IGSK, and the test set is essential to verify the efficiency of the final model of COVID-19 diagnosis.

The CNN model can also be developed by leveraging hyperparameter values from each IGSK solution. Regarding the performance of the CNN model for COVID-19 diagnosis, the fitness function can be interpreted. The images in the training set are considered for this purpose as inputs for the CNN model during the training process. The fitness value of solutions for optimization process is based on the accuracy of CNN classifier in classification of input images. The accuracy metric is the number of images classified successfully over the total number of images considered for the input set. The value of the accuracy criterion can thus be defined accordingly:

$$Accuracy = \frac{\#\text{ number of correctly classified images}}{\#\text{ total number of input images}} \tag{17}$$

A better accuracy solution necessarily has a higher fitness cost and likewise. The objective of the proposed approach is to further acquire solutions which comprise optimum values of CNN hyperparameters, that have the best accuracy rates (i.e. the best fitness cost values). This results in a high-performance CNN

model for classifying the images in the test set. The optimized CNN model is chosen to recognize the images in the test set after determining the appropriate values of CNN hyperparameters with IGSK.

2.3.3. Ensemble approach using reinforcement learning technique

Reinforcement learning is known as a powerful learning methodology focusing on the process of interaction with the environment. The main idea of this learning model is to find the optimal decision using trial and error leading to obtain the optimal solution. The Q-learning method is one of the most popular reinforcement learning algorithms which is based on updating Q values in the environment. Due to the effectiveness and good convergence of the Q-learning, we use this method to make ensemble classifier in the proposed method. In other words, the Q-learning method is employed to obtain an optimal subset of base classifiers to improve the classification accuracy of the proposed ensemble method.

Suppose $B_L = \{b_1, b_2, \ldots, b_L\}$ is the set of base CNN classifiers obtained by the Bagging method where L denotes the number of base classifiers. The main idea here is to select an optimal subset of the base classifiers denoted as $B'_{L'}$ using the reinforcement learning. To perform reinforcement learning, we need to define the states set S and the actions set A. In the proposed method, each state is represented as a tuple $s_t = [L_t, Acc_t]$ where L_t denotes the number of selected base CNN classifiers in the state s_t and Acc_t refers to the accuracy of the ensemble classifier constructed by the selected base CNN classifiers. Moreover, each action is defined as $a_t = \{0 \text{ or } 1\}$ where a base CNN classifier is randomly moved from B_L to $B'_{L'}$ when $a_t = 0$ and a base CNN classifier is randomly removed form $B'_{L'}$ when $a_t = 1$. In addition to the states set and actions set, we need to define a reward function for the Q-learning algorithm. To this end, we use the accuracy of the ensemble classifier. Suppose $y_i, i = 1, \ldots, n$ represents the input images that should be classified by a base CNN classifier and $b_l(y_i)$ is the output of the base classifier for the ith input sample. The value of $b_l(y_i)$ is determined as follows:

$$\mathrm{b}_l(y_i) = \begin{cases} 1, & \text{for COVID} \\ 0, & \text{for non-COVID cases} \end{cases} \tag{18}$$

To calculate the final output of the ensemble classifier, the outputs of the base CNN classifiers are combined based on the simple voting approach. Therefore, the final output of the ensemble classifier can be calculated as follows:

$$\mathbf{B}_l(y_i) = \begin{cases} 1, & \text{if } \sum_{l=1}^{L} \mathbf{b}_l(y_i) > L/2 \\ 0, & \text{otherwise} \end{cases} \tag{19}$$

It is worth noting that the purpose of the proposed method is to obtain an ensemble classifier with the highest accuracy by selecting an optimal subset of base classifiers using the Q-learning algorithm. To this end, the reward function used in the proposed method is defined as follows:

$$r_t = \begin{cases} 1, & Acc_t \geq Acc_o \\ 0, & Acc_t < Acc_o \end{cases} \tag{20}$$

where Acc_0 is the initial accuracy value of the ensemble classifier which is calculated based on the original set of classifiers B_L.

Reinforcement learning provides a mapping from each state to an action which is defined as the policy $\Pi : S \rightarrow A$. The aim of reinforcement learning is to find the optimal policy function leading to obtain the optimal subset of the base CNN classifiers. The Q-learning algorithm is based on an action-value function denoted as $Q^{\Pi}(s_t, a_t)$ where s_t and a_t are the state and action, respectively. To perform an action a under state s, the total expected discount reward value based on the optimal policy Π is calculated as follows:

$$Q^{\pi}(\mathbf{s}, a) = E\left(\sum_{k=0}^{\infty} \gamma^i r_t \mid \mathbf{s}_0 = \mathbf{s}, a_0 = a, \pi\right) \tag{21}$$

where γ refers to the discount coefficient which its value can be set as $0 \leq \gamma \leq 1$. The optimal action-value function $Q^*(s, a)$ can be calculated using the Q-learning algorithm as follows:

$$Q^*(\mathrm{s}, a) = E\left(r_{t+1} + \gamma \max_{a'} Q^*\left(\mathrm{s}_{t+1}, a'\right) \mid \mathrm{s}_t = \mathrm{s}, a_t = a\right) \tag{22}$$

In the proposed method, the Q-learning algorithm starts with initialization of action-value function Q with random values. Then, the algorithm continues a number of episodes where in each episode, the action-value function Q is updated to find an optimal policy for the search problem. To this end, the Q values are updated as follows:

$$Q(\mathbf{s}_t, a_t) \leftarrow Q(\mathbf{s}_t, a_t) + \alpha\left[r_{t+1} + \lambda \max_a Q(\mathbf{s}_{t+1}, a) - Q(\mathbf{s}_t, a_t)\right] \tag{23}$$

After performing the Q-learning algorithm, the optimal subset of the base CNN classifiers $B'_{L'}$ can be determined based on the best obtained result. Then, this optimal subset is used as the final base CNN classifiers to generate the proposed ensemble image classifier method. The general stages of the proposed COVID-19 diagnosis approach are depicted in Algorithm 1.

Algorithm 1 The pseudo-code of the proposed deep ensemble COVID-19 classification approach (DNE-RL)

Input: N (Population size), k_f (Knowledge factor), k_r (Knowledge ratio), GEN (Maximum number of generations), and L (Number of base classifiers).
Output: Detected COVID or Non-COVID images.
Begin algorithm:
Split images of dataset into training Tr and test Te sets;
Generate a set of base classifiers $B_L = \{b_1, b_2, \ldots, b_L\}$;
Set $m = 1$;
while $(m < L)$ **do**
  Generate a random initial population X_i (i=1, 2, …, N);
  Perform the OBL strategy to generate opposite solutions (ON);
  Set g=1;
  while $(g < GEN)$ **do**
    Set a CNN model for each solution in N and ON based on their hyperparameter values;
    Calculate the fitness of population $(N \cup ON)$ using Eq. (17) as the accuracy of CNN algorithm obtained by Tr set;
    Calculate the number of Gained and Shared dimensions of both phases using experience Eqs. (3) and (4);
    //Perform Junior gaining–sharing knowledge phase;
    //Perform Senior gaining–sharing knowledge phase;
    if fitness(x_i^{new}) ≤ fitness(x_i^{old}) **then**
      $x_i^{old} = x_i^{new}$, fitness(x_i^{old}) = fitness(x_i^{new})
    end if
    if fitness(x_i^{new}) ≤ fitness(x_{best}^{Global}) **then**
      $x_{best}^{Global} = x_i^{new}$, fitness(x_{best}^{Global}) = fitness(x_i^{new})
    end if
    Apply the CM operator based on Eq. (10);
    Set g=g+1;
  end while
  Consider the b_m base CNN classifier based on the hyperparameters obtained by the best search agent (x_{best}^{Global});
  Set m=m+1;
end while
Apply Q-learning algorithm on the base CNN classifiers B_L to select an optimal subset of the base classifiers as $B'_{L'}$;
Apply the proposed ensemble method to classify the images in the test set Te using the selected base classifiers $B'_{L'}$;
Return the classified images as the output;
End algorithm

Table 2
Involved hyperparameters in the evolutionary algorithm and their corresponding values.

Symbol	Value
K_s	[1, 30]
N_f	[1, 600]
Opt	[Adam, Adagrad, SGD, Adamax]
N_e	[1, 500]
B_s	[10, 20, . . . , 400]
N_c	[1, 2, . . . , 20]
MP_s	[1, 30]
D_r	[0.2, 0.25, . . . , 0.65]
Act	[ReLU, Sigmoid, Hard sigmoid, Tanh]
L_r	[0.001, 0.006, . . . , 0.1]
M_r	[0.05, 0.1, . . . , 0.95]

3. Experimental testing and analysis

3.1. Experimental setup

This section describes the configurations of the different parameters for the proposed method. For running the deep neural networks that are evolved by the proposed evolutionary algorithm, we utilized Keras which is a high-level framework for artificial neural networks with compatibility from TensorFlow. In regard to the IGSK evolutionary model parameterization, as we use one GeForce GTX 1080 Ti GPU and one 16 GB RAM, we have taken into account the time required to assess each individual that is, in particular, fairly high. We agreed to restrict the maximum number of iterations to 20 and the population size to 30 in order to alleviate this concern. These parameters have been chosen quantitatively to find a balance between the algorithm accuracy and execution time. Table 2 describes all hyperparameters as well as their integer or categorical values encoding in the deep learning algorithm. Nine hyperparameters including kernel size (K_s), number of filters (N_f), number of epochs (N_e), batch size (B_s), number of convolutional layers (N_c), dropout rate (D_r), maxpooling size (MP_s), learning rate (L_r), and momentum rate (M_r), are integer, whereas optimizer type (Opt) and activation function type (Act) are categorical hyperparameters that are optimized using the proposed IGSK to evolve the CNN architectures. In the proposed ensemble model, we use $L = 10$ base classifiers in which an optimal subset of them is selected through reinforcement learning. Also, the number of episodes in the Q-learning algorithm is set to 100. In the performed experiments, we chose the same values for the stochastic parameters that are common between the proposed method and other compared models in order to make a fair comparison. Also, for the other stochastic parameters used specifically in the compared models, we chose the optimal values reported in the corresponding papers. In addition, we perform a grid search and a trial and error procedure to choose the best values of the stochastic parameters used in different models.

3.2. COVID-19 X-ray image datasets

In this section, we introduce two well-known open-source image datasets named as Mendely and Kaggle which are the most commonly used datasets to analyze deep learning algorithms for COVID-19 diagnosis. In order to make a fair comparison between the proposed method and other benchmarks, we convert both datasets to 224*224 pixels which is a general rule for CNNs to get the same size. Furthermore, 80% of each dataset is considered for training purpose and the remaining images are utilized for testing. In Mendely dataset [65],[1] there are 1800 images, with 1510 images in normal cases, while the rest of the cases are influenced by COVID-19. The Kaggel dataset[2] also includes 1735 images, of which 825 images are related to cases with COVID-19 and the remaining 910 X-ray images are related to individuals who are not infected with COVID-19. To better understand the existing samples of individuals affected by COVID-19 and also to have normal conditions, we show some samples of X-ray images in both databases in Figs. 3 and 4 by their conditions.

3.3. Experimental results of benchmark functions

When introducing a new meta-heuristic, it is required to verify its performance before applying it on the considered case study. Benchmark objective functions are to be used to perform such benchmarking. In this study, 15 well-known benchmark objective functions serve this purpose as can be seen in Table 3. Besides, on the mentioned benchmarked functions, the proposed improved GSK (IGSK) in compared with the original version of GSK and eight powerful evolutionary algorithms in the literature including grasshopper optimization algorithm (GOA) [66], Slime mold algorithm (SMA) [67], genetic algorithm, gray wolf optimizer (GWO) [68], particle swarm optimization (PSO), differential evolution (DE), and biogeography-based optimization (BBO) [69].

Throughout the experiments conducted, the number of iterations and population sizes were established on 1000 and 40, respectively, as well as the dimension of the search space size was fixed to 30. The average (AVG) and standard deviation (STD) values of all algorithms are performed 40 times on 15 standard benchmark functions separately. The test of Wilcoxon sign rank is also utilized for the evaluation of IGSK's performance over 15 benchmark functions of all compared evolutionary algorithms.

Table 4 shows that not only IGSK remarkably improves the performance results and effectiveness of the standard GSK, but it further overcomes other approaches in the deployment of solutions throughout each benchmark function. Furthermore, all STDs in Table 4 indicate that IGSK has the smallest deviations between the eight benchmark methods. In other words, IGSK has better consistency and finds the excellent solutions in comparison with other approaches, which indicates that the proposed IGSK offers significant improvements that greatly surpass other powerful evolutionary competitors in fifteen bimodal, multimodal and hybrid functions.

In order to determine the significance between the proposed IGSK and other eight benchmarked algorithms, the Wilcoxon sign rank test is applied on the 15 benchmark functions as its detailed results can be seen in Table 5. Based on this table, if the p-value of Wilcoxon test is less than 0.05, it means that the IGSK algorithm provides significant advantages against all of its competitors. The results show that all of the p-values for the eight compared algorithms are substantially lower than 0.05 in 15 functions indicating the dominance of IGSK over all benchmarked algorithms.

3.4. Experimental results of COVID-19 datasets

In this section, we compare the competencies and the superiority of our proposed DNE-RL model to a comprehensive comparison of various evolutionary algorithms and well-known deep learning architectural models. Four common techniques of optimization other than IGSK are used in this section for image classification. These algorithms include GA (genetic algorithm) [27], DE (differential evolution) [70], PSO (particle swarm optimization) [71], and the original GSK. MobileNet, VGGNet19,

[1] https://data.mendeley.com/datasets/2fxz4px6d8/4

[2] https://www.kaggle.com/tawsifurrahman/covid19-radiography-database

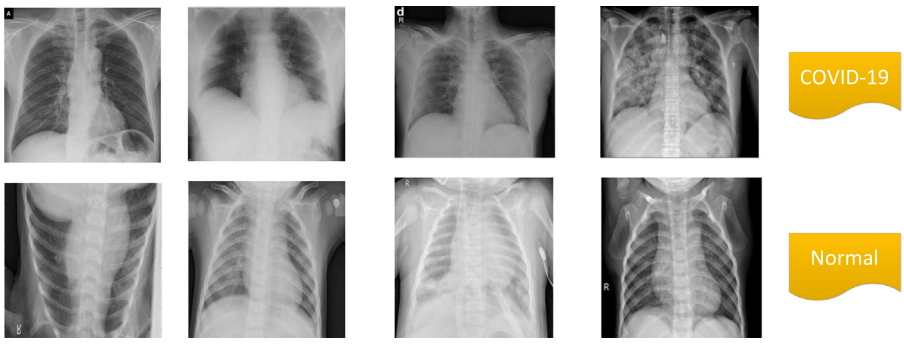


Fig. 3. Four Samples of the X-ray-based Mendely dataset.

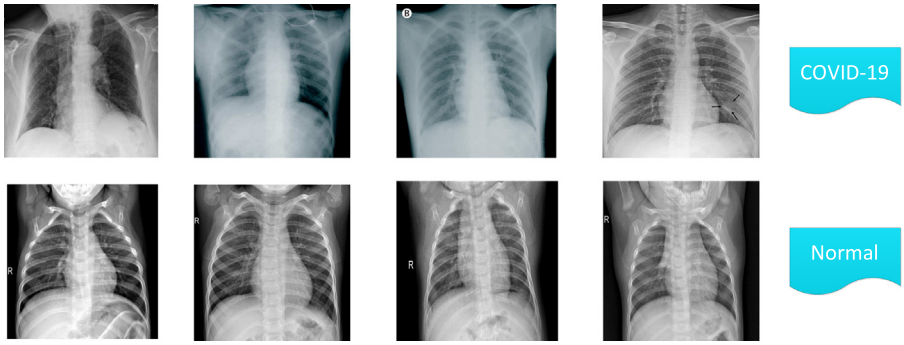


Fig. 4. Four Samples of the X-ray-based Kaggle dataset.

Table 3
Description of the 15 benchmark functions.

No.	Functions	Search range	$F_i^* = F_i(x^*)$
Unimodal functions			
F1	Rotated High Conditioned Elliptic Function	[−100, 100]	100
F2	Rotated Bent Cigar Function	[−100, 100]	200
F3	Rotated Discus Function	[−100, 100]	300
Multimodal functions			
F4	Shifted and Rotated Rosenbrock's Function	[−100, 100]	400
F5	Shifted and Rotated Ackley's Function	[−100, 100]	500
F6	Shifted and Rotated Schwefel's Function	[−100, 100]	1100
F7	Shifted and Rotated Katsuura Function	[−100, 100]	1200
F8	Shifted and Rotated HappyCat Function	[−100, 100]	1300
F9	Shifted and Rotated HGBat Function	[−100, 100]	1400
F10	Shifted and Rotated Expanded Griewank's plus Rosenbrock's Function	[−100, 100]	1500
Hybrid functions			
F11	Hybrid Function 1 (N = 3)	[−100, 100]	1700
F12	Hybrid Function 2 (N = 3)	[−100, 100]	1800
F13	Hybrid Function 3 (N = 4)	[−100, 100]	1900
F14	Hybrid Function 4 (N = 4)	[−100, 100]	2000
F15	Hybrid Function 5 (N = 5)	[−100, 100]	2100

ResNet50, and DenseNet12 are four powerful deep learning architectures that are extensively used as the benchmarks in image classification works in the literature. It is worth mentioning that the GA, DE, PSO, and GSK models are ensemble classification approaches used in the experiments to evaluate the performance of the proposed evolutionary model. Also, the MobileNet, VGGNet19, ResNet50, and DenseNet12 models are used in the experiments as single classification approaches to show this issue that how the proposed ensemble model can provide better performance than the single models. 75% of each dataset is used for training the deep learning networks; while the remaining 25% is used for test. The accuracy, precision, recall, F-measure, and AUC as the performance classification metrics of all these optimized and non-optimized deep architectural models for Mendely and Kaggle datasets are reported in Tables 6 and 7.

Table 6 shows the results of our proposed DNE-RL algorithm and the other eight competing algorithms based on five classification indicators for Mendely dataset. The results of this table show that the proposed DNE-RL has a significant advantage over other comparable algorithms based on the performance metrics. For example, based on the average accuracy metric, the algorithm proposed by us has about two percent superior to its nearest follower (VGGNet19). The superiority of the proposed algorithm with its closest competing algorithm (VGGNet19) is about one percent, two percent, two percent, and one and a half percent, respectively, based on the average of precision, recall, F-measure, and AUC indices. Also, the performance results of the compared algorithms with the proposed algorithm for Kaggle dataset are shown in Table 7. These results demonstrate that our proposed

Table 4
The experimental results for the proposed IGSK compared with well-known evolutionary algorithms.

Function	Statistic	GOA	SMA	GA	GWO	PSO	DE	BBO	GSK	IGSK
F1	AVG	9.59E+07	3.18E+08	1.70E+08	1.06E+08	1.79E+07	1.19E+07	4.21E+08	6.43E+08	**8.10E+06**
	STD	6.35E+07	1.75E+08	4.23E+07	8.36E+07	4.10E+06	8.27E+06	1.33E+08	1.29E+08	**3.11E+06**
F2	AVG	2.90E+09	7.72E+09	2.59E+09	1.39E+10	1.93E+08	2.46E+07	2.79E+10	4.96E+10	**1.39E+04**
	STD	1.66E+09	3.35E+09	1.74E+09	6.93E+09	2.28E+07	3.30E+06	6.16E+09	3.78E+09	**7.67E+03**
F3	AVG	3.90E+04	1.82E+05	8.39E+04	9.46E+04	3.66E+04	1.12E+05	6.09E+04	1.38E+05	**7.82E+03**
	STD	1.24E+04	4.28E+04	2.89E+04	7.02E+04	7.55E+03	3.06E+04	1.07E+04	2.71E+04	**6.59E+03**
F4	AVG	6.89E+02	1.37E+03	8.98E+02	1.53E+03	5.07E+02	5.12E+02	2.57E+03	5.66E+03	**6.81E+02**
	STD	1.15E+02	3.88E+02	1.92E+02	1.01E+03	3.93E+01	3.55E+01	7.82E+02	9.43E+02	**4.83E+01**
F5	AVG	5.17E+02	5.22E+02	5.28E+02	5.44E+02	5.38E+02	5.19E+02	5.42E+02	5.26E+02	**5.14E+02**
	STD	5.80E−02	5.67E−02	8.83E−02	9.77E−02	4.85E−02	7.11E−02	6.76E−02	4.85E−02	**3.14E−02**
F6	AVG	4.42E+03	7.33E+03	7.57E+03	6.33E+03	6.55E+03	5.81E+03	8.75E+03	8.55E+03	**2.23E+03**
	STD	1.47E+03	6.39E+02	8.15E+02	7.34E+02	6.41E+02	6.63E+02	4.89E+02	3.27E+02	**4.88E+02**
F7	AVG	1.15E+03	1.15E+03	1.15E+03	1.14E+03	1.15E+03	1.15E+03	1.15E+03	1.15E+03	**1.14E+03**
	STD	1.17E+00	5.61E−01	6.59E−01	2.18E−01	4.08E−01	3.11E−01	3.72E−01	3.31E−01	**8.18E−02**
F8	AVG	1.27E+03	1.28E+03	1.27E+03	1.27E+03	1.27E+03	1.27E+03	1.28E+03	1.27E+03	**1.26E+03**
	STD	4.08E−01	8.96E−01	9.72E−02	1.24E+00	7.40E−02	7.72E−02	2.63E−01	3.76E−01	**6.73E−02**
F9	AVG	1.37E+03	1.40E+03	1.38E+03	1.39E+03	1.38E+03	1.38E+03	1.44E+03	1.50E+03	**1.36E+03**
	STD	6.94E+00	1.09E+01	4.85E+00	1.93E+01	1.66E−01	1.45E−01	1.56E+01	1.66E+01	**5.11E−02**
F10	AVG	1.71E+03	8.24E+03	1.51E+03	2.44E+05	1.39E+03	1.39E+03	1.64E+04	1.41E+05	**1.39E+03**
	STD	6.88E+02	1.45E+03	6.92E+01	1.09E+05	9.88E+01	2.97E+01	1.13E+04	5.89E+04	**7.21E+00**
F11	AVG	2.23E+06	1.29E+07	1.63E+07	4.25E+06	7.23E+05	5.64E+05	1.52E+07	2.27E+07	**1.22E+06**
	STD	1.78E+06	1.03E+07	7.21E+06	1.18E+06	5.31E+05	3.23E+05	7.31E+06	8.03E+06	**1.67E+05**
F12	AVG	7.39E+06	7.73E+06	5.07E+05	1.03E+08	2.07E+06	2.89E+05	3.07E+08	1.14E+09	**4.71E+03**
	STD	2.95E+06	3.70E+06	9.29E+05	3.36E+08	8.88E+05	2.03E+04	1.93E+08	4.87E+08	**2.88E+03**
F13	AVG	1.92E+03	1.97E+03	1.97E+03	1.95E+03	1.90E+03	1.89E+03	2.00E+03	2.14E+03	**1.78E+03**
	STD	2.80E+01	5.09E+01	3.87E+01	5.27E+01	1.08E+01	1.30E+01	3.85E+01	3.85E+01	**9.55E+00**
F14	AVG	3.04E+04	1.05E+05	8.36E+04	6.99E+04	1.67E+04	2.68E+04	4.00E+04	8.73E+04	**1.21E+04**
	STD	1.34E+04	7.17E+04	6.26E+04	5.65E+04	6.51E+03	1.46E+04	1.87E+04	3.38E+04	**6.08E+03**
F15	AVG	1.91E+06	5.69E+06	1.36E+07	8.93E+05	6.02E+06	2.58E+05	3.49E+06	4.99E+06	**2.34E+05**
	STD	1.15E+06	3.84E+06	8.78E+06	1.55E+05	4.23E+06	2.18E+05	2.18E+06	1.92E+06	**1.90E+05**

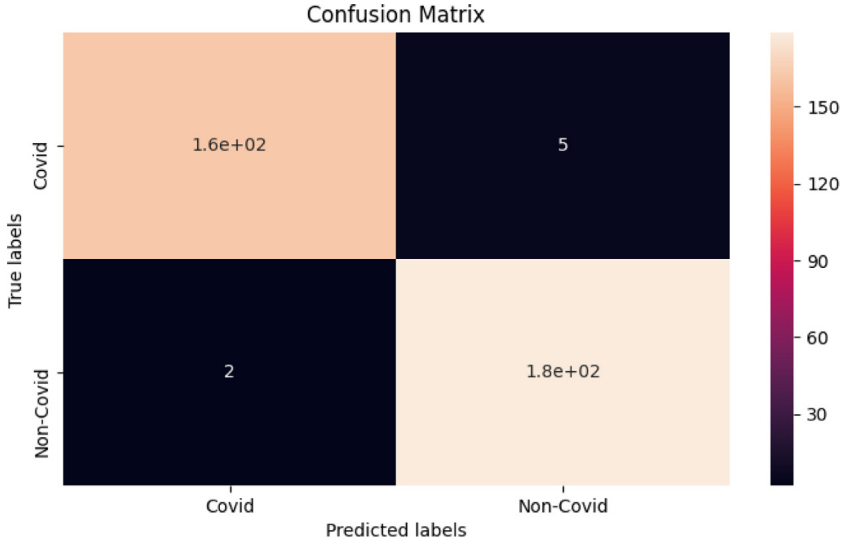
Table 5
The p-values of Wilcoxon test between the proposed IGSK and other benchmark evolutionary algorithms.

Function	GOA	SMA	GA	GWO	PSO	DE	BBO	GSK
F1	1.65E−06	1.65E−06	2.88E−06	2.88E−06	1.24E−01	2.93E−01	1.65E−06	1.65E−06
F2	1.65E−06	1.65E−06	1.65E−06	1.65E−06	1.65E−06	1.65E−06	1.65E−06	1.65E−06
F3	2.44E−06	1.65E−06	1.65E−06	4.78E−06	1.65E−06	1.65E−06	1.65E−06	1.65E−06
F4	1.65E−06	1.65E−06	1.65E−06	2.18E−06	5.88E−04	2.67E−02	1.65E−06	1.65E−06
F5	1.65E−06	1.65E−06	1.65E−06	6.64E−04	1.65E−06	1.65E−06	1.65E−06	1.65E−06
F6	3.43E−05	1.65E−06	1.65E−06	1.65E−06	1.65E−06	1.65E−06	1.65E−06	1.65E−06
F7	3.33E−06	1.65E−06	1.65E−06	2.68E−03	1.65E−06	1.65E−06	1.65E−06	1.65E−06
F8	2.24E−03	1.65E−06	1.99E−03	1.79E−06	4.01E−03	1.73E−03	1.65E−06	4.88E−06
F9	1.99E−05	1.65E−06	5.52E−04	1.65E−06	3.57E−04	1.82E−02	1.65E−06	2.08E−06
F10	2.09E−05	1.65E−06	1.65E−06	1.65E−06	3.49E−02	1.53E−02	1.65E−06	1.65E−06
F11	4.11E−02	1.65E−06	1.65E−06	4.23E−03	2.08E−05	2.56E−06	2.78E−06	7.73E−06
F12	1.65E−06	1.65E−06	1.65E−06	8.45E−05	1.65E−06	1.65E−06	1.65E−06	1.65E−06
F13	8.77E−05	1.65E−06	2.58E−06	7.03E−06	3.81E−04	5.34E−04	1.65E−06	1.65E−06
F14	4.51E−06	1.96E−06	1.65E−06	1.87E−06	1.24E−02	5.19E−05	3.22E−06	2.01E−06
F15	2.76E−05	1.93E−05	1.65E−06	4.11E−04	4.16E−03	3.19E−03	1.65E−06	2.14E−05

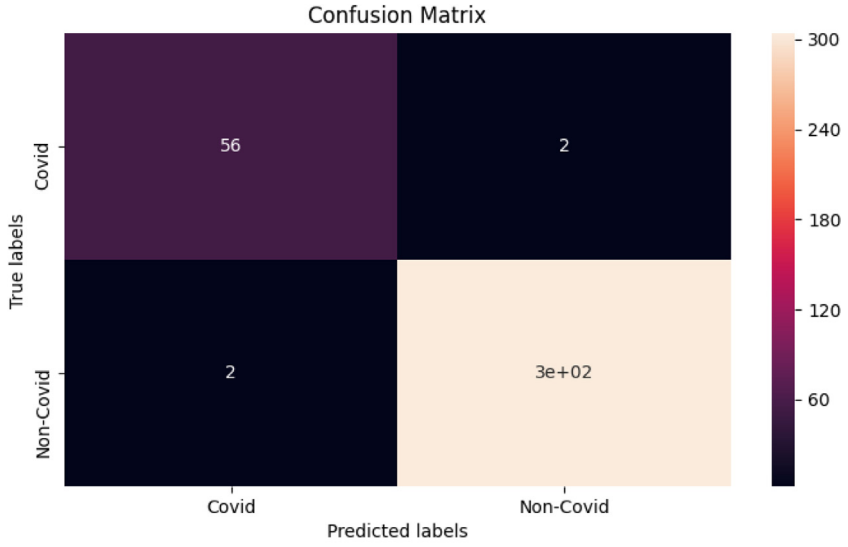
algorithm performs best in five performance metrics compared to other benchmark algorithms for COVID-19 diagnosis.

The confusion matrix is one of the viable strategies to determine the effectiveness of classification algorithms. This technique works by dividing input data samples according to true/false predictions in four sub-sets. Fig. 5 shows the confusion matrices created by our proposed DNE-RL algorithm for both Mendely and Kaggle datasets. The confusion matrix of the proposed DNE-RL framework, as can be seen in this figure, gives more accurate results than the other deep learning algorithms. It is important to realize that the best possible outcome is a confusion matrix containing more true predicted observations. It can therefore be inferred that the proposed method has better performance than other classification algorithms.

Box plot is another important tool for measuring the performance of algorithms from the statistical point of view in machine learning topics, which is based on four indicators: best, worst, average and standard deviation values of each classification model. To this end, the performance of DNE-RL model and other competing deep models for the two COVID-19 datasets based on fitness function (accuracy metric) are shown in Figs. 6 and 7. These figures indicate that, in obtaining the best average accuracy, the proposed DNE-RL framework is considered as the most efficient model among all deep classification models for both Mendely and Kaggle datasets. Another conclusion to be drawn from these figures is that for the Mendely dataset, the DenseNet12 algorithm, and for the Kaggle dataset, the ResNet50 algorithm, both of which are non-optimized algorithms, are the closest competitors to the proposed DNE-RL algorithm which show the acceptable performance. Among the four search algorithms (GA, DE, PSO and GSK), the original version of GSK performs better than the rest of the search algorithms.



(a) Kaggle dataset.



(b) Mendely dataset.

Fig. 5. Confusion matrices of the proposed DNE-RL model for Mendely and Kaggle datasets.

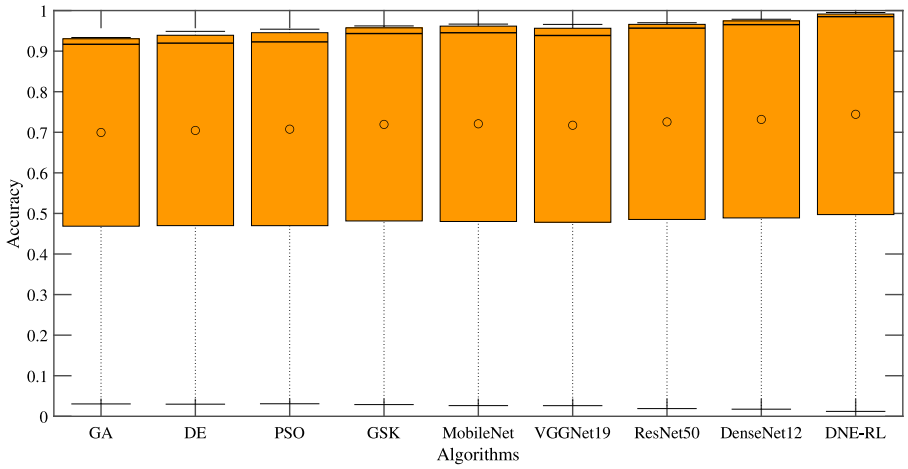


Fig. 6. Box plots of the proposed and other benchmark models for Mendely dataset.

Table 6
The experimental performance of the proposed model (DNE-RL) vs other competitive deep learning benchmarks for Mendely Dataset.

Metric		GA	DE	PSO	GSK	MobileNet	VGGNet19	ResNet50	DenseNet12	DNE-RL
ACC	AVG	0.927389	0.929467	0.936799	0.952668	0.956657	0.946883	0.961993	0.970338	0.987742
	STD	0.030495	0.029899	0.030885	0.028858	0.026337	0.026199	0.018996	0.017557	0.012116
	Best	0.933583	0.948779	0.954091	0.962122	0.966641	0.965887	0.969985	0.978594	0.995186
	Worst	0.906488	0.909957	0.908777	0.934014	0.933933	0.930258	0.951332	0.960241	0.982168
Precision	AVG	0.942783	0.938861	0.947751	0.948183	0.963662	0.956788	0.968848	0.973861	0.984334
	STD	0.025571	0.024167	0.023885	0.023955	0.021175	0.035537	0.023191	0.022818	0.013883
	Best	0.956733	0.957883	0.959919	0.959192	0.975455	0.966881	0.976766	0.978588	0.991443
	Worst	0.916855	0.917919	0.920559	0.926634	0.939986	0.922065	0.941445	0.947766	0.984452
Recall	AVG	0.921456	0.918872	0.925531	0.959983	0.948711	0.938636	0.956674	0.962678	0.989123
	STD	0.009886	0.009817	0.013553	0.011762	0.012835	0.011028	0.014224	0.016887	0.009931
	Best	0.931441	0.923221	0.936637	0.963321	0.957766	0.945709	0.966838	0.968871	0.992172
	Worst	0.918965	0.914453	0.921298	0.949663	0.931655	0.925667	0.944913	0.948815	0.985561
F-measure	AVG	0.924881	0.927571	0.932889	0.948775	0.954686	0.943008	0.957814	0.968649	0.984939
	STD	0.002882	0.003662	0.003119	0.005891	0.018893	0.015944	0.019962	0.004881	0.002674
	Best	0.927644	0.931089	0.938875	0.095219	0.961008	0.949889	0.961129	0.971291	0.987175
	Worst	0.920307	0.922566	0.926618	0.944103	0.948871	0.939892	0.939072	0.965667	0.982323
AUC	AVG	0.929331	0.928888	0.937676	0.951881	0.957117	0.945771	0.963442	0.969881	0.988466
	STD	0.017553	0.024554	0.015585	0.008914	0.005771	0.023319	0.019288	0.024596	0.015884
	Best	0.932441	0.937669	0.942448	0.955676	0.960083	0.957119	0.972885	0.982011	0.991777
	Worst	0.925669	0.917765	0.932191	0.048911	0.953382	0.928228	0.948558	0.939006	0.981407

Table 7
The experimental performance of the proposed model (DNE-RL) vs other competitive deep learning benchmarks for Kaggle dataset.

Metric		GA	DE	PSO	GSK	MobileNet	VGGNet19	ResNet50	DenseNet12	DNE-RL
ACC	AVG	0.933488	0.924888	0.942884	0.948021	0.949502	0.950001	0.972881	0.966913	0.991441
	STD	0.027901	0.012288	0.011142	0.014252	0.017881	0.023315	0.020994	0.013637	0.011042
	Best	0.947991	0.937786	0.948788	0.958893	0.958812	0.961152	0.982441	0.971189	0.993668
	Worst	0.912298	0.915658	0.931229	0.932285	0.937881	0.927748	0.955571	0.952449	0.980671
Precision	AVG	0.949893	0.940174	0.956618	0.957782	0.958838	0.968841	0.982441	0.973861	0.993568
	STD	0.037716	0.020781	0.030083	0.025151	0.023878	0.022274	0.027313	0.019669	0.021339
	Best	0.968668	0.962881	0.972335	0.973991	0.968816	0.975561	0.986004	0.983315	0.996812
	Worst	0.908915	0.921007	0.913806	0.931401	0.927175	0.944767	0.952721	0.955672	0.968814
Recall	AVG	0.925933	0.912829	0.935581	0.929181	0.935535	0.942552	0.960112	0.953443	0.981445
	STD	0.014047	0.018626	0.007996	0.009158	0.014663	0.018596	0.010052	0.008492	0.011381
	Best	0.940116	0.936682	0.941887	0.937736	0.948778	0.955778	0.968784	0.959928	0.987886
	Worst	0.913005	0.904334	0.929981	0.920704	0.923051	0.932994	0.953008	0.942217	0.970221
F-measure	AVG	0.932717	0.921633	0.940081	0.945771	0.947006	0.948003	0.969951	0.964412	0.989666
	STD	0.015542	0.008797	0.012252	0.002363	0.006666	0.011882	0.016672	0.012331	0.004881
	Best	0.938812	0.935542	0.948788	0.949772	0.951332	0.954481	0.976866	0.969982	0.990454
	Worst	0.920094	0.916652	0.931176	0.938182	0.942553	0.935991	0.951331	0.957871	0.983353
AUC	AVG	0.934451	0.926656	0.940092	0.949882	0.951666	0.947781	0.974554	0.963999	0.990337
	STD	0.022444	0.011434	0.008892	0.006772	0.021033	0.014403	0.009037	0.014544	0.007881
	Best	0.945333	0.929922	0.948871	0.951991	0.969912	0.952662	0.979988	0.970002	0.992008
	Worst	0.911329	0.915004	0.934505	0.943303	0.936766	0.932242	0.968998	0.949889	0.982662

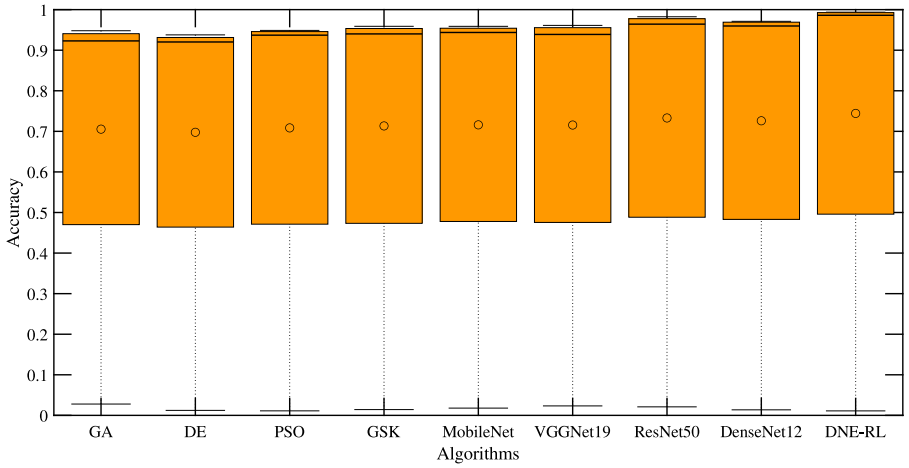


Fig. 7. Box plots of the proposed and other benchmark models for Kaggle dataset.

Figs. 8 and 9 demonstrate the convergence curves of the proposed method based on the training set and test set for Mendely and Kaggle datasets, respectively. These figures are helpful to show the trend of the proposed method's convergence on both

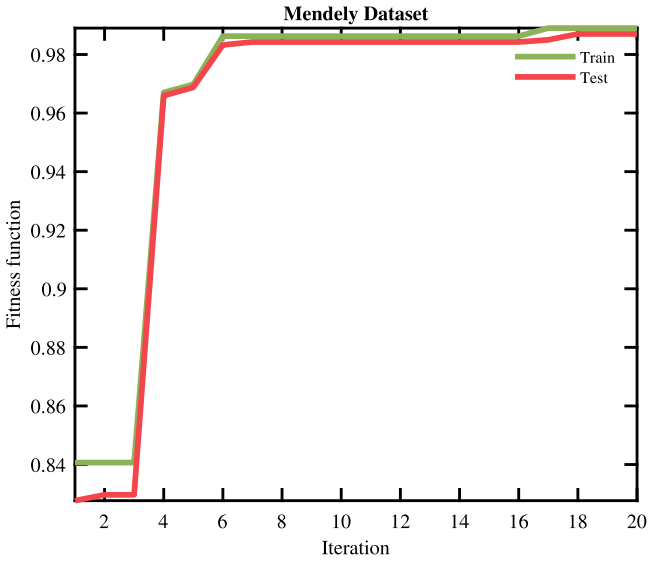


Fig. 8. Convergence curves of the proposed method based on the training set and test set for Mendely dataset.

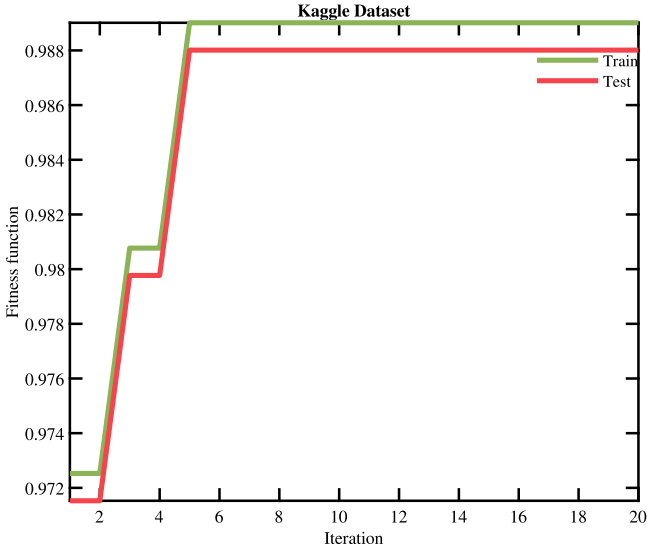


Fig. 9. Convergence curves of the proposed method based on the training set and test set for Kaggle dataset.

the training and test sets in different iterations. Also, the analysis of the over-fitting and under-fitting problems can be inferred form these figures. It is worth mentioning that a model that faces with the under-fitting problem will have high training and high testing error. On the other hand, a model with an over-fitting problem will have extremely low training error but a high testing error. By investigating the convergence curves of the proposed method demonstrated in Figs. 8 and 9, we can find that the performance of the proposed method on the training set is very close to the test set in different iterations. Therefore, it can be concluded that the proposed method performs normally without any over-fitting and/or under-fitting problems.

For the evaluation and interpretation of the performance of the proposed model and all benchmark algorithms, two statistical tests are implemented with the use of the T-test and Friedman ranking tests. Table 8 shows the p-values obtained from comparing the proposed DNE-RL framework with other benchmark algorithms on both Mendely and Kaggle datasets. The results of this statistical test show that our proposed algorithm is significantly different from other comparable algorithms. Also, in Tables 9 and 10, the performance metric results of the Friedman ranking test for all algorithms applied to the Mendely and Kaggle datasets are reported. In Table 9, which is related to the ranking results of all algorithms applied to Mendely dataset, the algorithm proposed by us ranks first, and the other two followers of this algorithm are the DenseNet12 and ResNet50 models, which are recognized as the well-known deep learning models in image processing tasks. Also, as we can see from Table 10, the ranking results on Kaggle dataset show that DNE-RL framework still has the best ranking among all the compared algorithms in terms of all performance metrics. ResNet50 and DenseNet12 as pre-trained deep learning models also rank second and third, respectively. The interesting point among the optimized algorithms is that the best framework optimized by the original version of GSK evolutionary ranks fifth among all competitors in the Kaggle dataset. This indicates that the use of optimization algorithms does not necessarily show the best performance. In addition, as our results show, the well-known pre-learning algorithms are the closest competitors to our proposed algorithm. One of the main reasons for the superiority of our algorithm over other competitors is that we have applied two strong optimization operators including OBL and Cauchy mutation strategies on the GSK algorithm, which increase the gain on most of the exploration and exploitation phases to achieve the optimal solution faster. Besides, in the proposed ensemble method, the reinforcement learning is employed to select an optimal subset of the base optimized CNN classifiers. The final output for the proposed classification method is obtained by integrating the outputs of the selected base classifiers. The results of experiments efficiently show that how the proposed ensemble method can outperform other classification models based on the used datasets and different evaluation metrics.

3.5. Run time experiments

In this section, we investigate the performance of the proposed algorithm and other compared methods on both the Mendely and Kaggle datasets in terms of execution time consumed by different algorithms. We consider the execution time of algorithms from three time perspectives such as optimization time, training time, and test time. It should be noted that the optimization time is not reported for the MobileNet, VGGNet19, ResNet50, and DenseNet12 models because they do not perform any optimization process in their procedures. As can be seen from Tables 11 and 12, our proposed algorithm takes the least amount of time in terms of three time indicators among all the compared algorithms. An interesting finding from these results is that other optimized algorithms, such as GA, DE, PSO and GSK take less training and test time than non-optimized algorithms, which underscores the importance of optimized algorithms in deep neural networks.

4. Discussion

This section discusses the strengths and weaknesses of the proposed method in detail. To this end, we can consider three main advantages for the proposed method. First, some appropriate modifications are made in the original version of the GSK evolutionary algorithm to improve its search ability to faster obtain the best possible solutions and reduce the probability of falling into local optima. To evaluate the efficiency of the proposed IGSK evolutionary algorithm, several experiments are conducted on 15 benchmark functions where the results demonstrate that the proposed algorithm performs better than other

Table 8
p-values of the proposed model vs other deep learning benchmarks for Mendely and Kaggle datasets.

Dataset	GA	DE	PSO	GSK	MobileNet	VGGNet19	ResNet50	DenseNet12
Mendely	3.84E−09	4.72E−04	1.69E−03	2.06E−03	2.32E−03	5.94E−04	9.83E−04	9.12E−04
Kaggle	7.71E−05	1.27E−05	7.48E−05	2.50E−04	1.21E−04	8.94E−04	9.07E−04	1.08E−03

Table 9
The Friedman test ranking results for the proposed framework and other deep learning algorithms for Mendely dataset using various performance metrics.

Metric	GA	DE	PSO	GSK	MobileNet	VGGNet19	ResNet50	DenseNet12	DNE-RL
ACC	9	8	7	5	4	6	3	2	1
Precision	8	9	7	6	4	5	3	2	1
Recall	8	9	7	3	5	6	4	2	1
F-measure	9	8	7	5	4	6	3	2	1
AUC	8	9	7	5	4	6	3	2	1
Summation	42	43	35	24	21	29	16	10	5
Average	8.4	8.6	7	4.8	4.2	5.8	3.2	2	1
Final Ranking	8	9	7	5	4	6	3	2	1

Table 10
The Friedman test ranking results for the proposed framework and other deep learning algorithms for Kaggle dataset using various performance metrics.

Metric	GA	DE	PSO	GSK	MobileNet	VGGNet19	ResNet50	DenseNet12	DNE-RL
ACC	8	9	7	6	5	4	2	3	1
Precision	8	9	7	6	5	4	2	3	1
Recall	8	9	5	7	6	4	2	3	1
F-measure	8	9	7	6	5	4	2	3	1
AUC	8	9	7	5	4	6	2	3	1
Summation	40	45	33	30	25	22	10	15	5
Average	8	9	6.6	6	5	4.4	2	3	1
Final Ranking	8	9	7	6	5	4	2	3	1

Table 11
Run-time comparison of different compared models based on Mendely dataset.

Model	Optimization time	Training time	Test time
MobileNet	–	734	253
VGGNet19	–	645	236
ResNet50	–	712	244
DenseNet12	–	634	228
GA	4516	531	216
DE	4431	517	191
PSO	3823	496	173
GSK	3711	473	158
DNE-RL	3609	432	146

Table 12
Run-time comparison of different compared models based on Kaggle dataset.

Model	Optimization time	Training time	Test time
MobileNet	–	664	239
VGGNet19	–	571	225
ResNet50	–	629	231
DenseNet12	–	553	216
GA	4129	487	187
DE	4037	461	169
PSO	3496	435	151
GSK	3409	421	139
DNE-RL	3312	386	124

compared evolutionary algorithms. Second, the proposed IGSK evolutionary algorithm is applied to deep CNN models to achieve the optimal values of their hyperparameters. This leads to significantly improve the performance of the deep CNN models in classifying the COVID-19 images. Different from other traditional deep CNN based image classifiers that determine the values of hyperparameters manually through trial and error, the proposed method automatically obtains the optimal values of the hyperparameters. Third, we propose an ensemble image classifier based on deep RL model. In particular, an optimal subset of CNN classifiers is obtained using the deep RL model. Then, the final classification output is determined by integrating the results of these selected CNN classifiers. Different to the traditional ensemble strategies that are focused on the trial and error procedures to obtain the best solutions which is a time-consuming task, the proposed ensemble model obtains the best combination of possible ensembles by the deep RL model. On the other hand, the deep RL ensemble strategy reduces the size of initial set of CNN classifiers by obtaining an optimal subset of them leading to reduce the computational costs raised by traditional ensemble approaches and also make an excellent classification performance. To the best of our knowledge, the proposed method is the first one which applies the concepts of evolutionary algorithm, deep CNN model, and deep reinforcement learning model to make an ensemble image classification method for the COVID-19 diagnosis. The effectiveness of the proposed method is evaluated on two well-known COVID-19 X-ray image datasets and the results show the superiority of the proposed method in comparison to other image classification models. Therefore, the proposed method is applicable to utilize in the hospitals to diagnose the infected COVID-19 and normal individuals. One of the main weaknesses of the proposed method is its relatively high run time. This is mainly due to performing the proposed IGSK evolutionary algorithm and the deep RL model to obtain the best possible solutions. This weakness can be alleviated by developing parallel algorithms for the search process of the proposed method. Another weakness of the proposed method is to ignore an effective preprocessing approach to apply on the input images to improve their quality. This issue can significantly enhance the accuracy of the used image classification models. These weaknesses can be addressed in the future works.

5. Conclusion

In this paper, we proposed a novel deep ensemble image classification model to detect COVID-19 cases based on chest X-ray

images. To this end, we employed deep CNN model as the base image classifier approach. It is worth noting that the performance of the CNN model significantly depends on the way its hyperparameters are adjusted. Therefore, we developed an improved version of GSK optimization algorithm and applied it to obtain the optimal values of the CNN's hyperparameters. Moreover, the reinforcement learning is used in the proposed method to make an ensemble classifier by selecting an optimal subset of the base CNN classifiers. The outputs of the selected base classifiers are combined to generate the final output of the proposed ensemble approach. The performed experiments on two well-known datasets demonstrated that the proposed method significantly outperforms other image classification methods. Therefore, it can be concluded that the proposed method can be employed as a powerful tool to detect COVID-19 cases. For the future works, we can employ other evolutionary algorithms to optimize the hyperparameters of the deep neural networks. Also, other deep classification models can be applied as the base classifiers in the proposed method.

CRediT authorship contribution statement

Seyed Mohammad Jafar Jalali: Conceptualization, Methodology, Software, Writing - original draft. **Milad Ahmadian:** Writing - original draft. **Sajad Ahmadian:** Conceptualization, Methodology, Writing - original draft, Writing - review & editing. **Abbas Khosravi:** Writing - original draft, Project administration. **Mamoun Alazab:** Writing - review & editing, Project administration. **Saeid Nahavandi:** Project administration.

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.

References

[1] C.-C. Lai, T.-P. Shih, W.-C. Ko, H.-J. Tang, P.-R. Hsueh, Severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) and corona virus disease-2019 (COVID-19): The epidemic and the challenges, Int. J. Antimicro. Ag. (2020) 105924.

[2] Y. Dong, X. Mo, Y. Hu, X. Qi, F. Jiang, Z. Jiang, S. Tong, Epidemiological characteristics of 2143 pediatric patients with 2019 coronavirus disease in China, Pediatrics (2020).

[3] M. Vaduganathan, O. Vardeny, T. Michel, J.J. McMurray, M.A. Pfeffer, S.D. Solomon, Renin–Angiotensin–Aldosterone system inhibitors in patients with covid-19, N. Engl. J. Med. 382 (17) (2020) 1653–1659.

[4] M.A. Shereen, S. Khan, A. Kazmi, N. Bashir, R. Siddique, COVID-19 infection: Origin, transmission, and characteristics of human coronaviruses, J. Adv. Res. (2020).

[5] M.P. Lythgoe, P. Middleton, Ongoing clinical trials for the management of the COVID-19 pandemic, Trends Pharmacol. Sci. (2020).

[6] V. Khosrawipour, H. Lau, T. Khosrawipour, P. Kocbach, H. Ichii, J. Bania, A. Mikolajczyk, Failure in initial stage containment of global COVID-19 epicenters, J. Med. Virol. (2020).

[7] E. Shim, A. Tariq, W. Choi, Y. Lee, G. Chowell, Transmission potential and severity of COVID-19 in South Korea, Int. J. Infect. Dis. (2020).

[8] P. Bahl, C. Doolan, C. de Silva, A.A. Chughtai, L. Bourouiba, C.R. MacIntyre, Airborne or droplet precautions for health workers treating COVID-19? J. Infect. Dis. (2020).

[9] T. Chen, S. Guo, P. Zhong, Epidemic characteristics of the COVID-19 outbreak in tianjin, a well-developed city in China, Am. J. Infect. Control (2020).

[10] M.H. Nasirpour, A. Sharifi, M. Ahmadi, S.J. Ghoushchi, Revealing the relationship between solar activity and COVID-19 and forecasting of possible future viruses using multi-step autoregression (MSAR), Environ. Sci. Pollut. Res. (2021) 1–11.

[11] M. Ahmadi, A. Sharifi, S. Khalili, Presentation of a developed sub-epidemic model for estimation of the COVID-19 pandemic and assessment of travel-related risks in Iran, Environ. Sci. Pollut. Res. 28 (12) (2021) 14521–14529.

[12] X. Zhang, D. Wang, Z. Zhou, Y. Ma, Robust low-rank tensor recovery with rectification and alignment, IEEE Trans. Pattern Anal. Mach. Intell. 43 (1) (2019) 238–255.

[13] X. Zhang, M. Fan, D. Wang, P. Zhou, D. Tao, Top-k feature selection framework using robust 0-1 integer programming, IEEE Trans. Neural Netw. Learn. Syst. (2020).

[14] X. Zhang, T. Wang, J. Wang, G. Tang, L. Zhao, Pyramid channel-based feature attention network for image dehazing, Comput. Vis. Image Underst. 197 (2020) 103003.

[15] X. Zhang, J. Wang, T. Wang, R. Jiang, J. Xu, L. Zhao, Robust feature learning for adversarial defense via hierarchical feature alignment, Inform. Sci. (2020).

[16] F. Tahmasebi, M. Meghdadi, S. Ahmadian, K. Valiallahi, A hybrid recommendation system based on profile expansion technique to alleviate cold start problem, Multimedia Tools Appl. 80 (2) (2021) 2339–2354.

[17] S. Ahmadian, M. Meghdadi, M. Afsharchi, Incorporating reliable virtual ratings into social recommendation systems, Appl. Intell. 48 (11) (2018) 4448–4469.

[18] S. Ahmadian, P. Moradi, F. Akhlaghian, An improved model of trust-aware recommender systems using reliability measurements, in: 2014 6th Conference on Information and Knowledge Technology, IKT, IEEE, 2014, pp. 98–103.

[19] P. Moradi, F. Rezaimehr, S. Ahmadian, M. Jalili, A trust-aware recommender algorithm based on users overlapping community structure, in: 2016 Sixteenth International Conference on Advances in ICT for Emerging Regions, ICTer, IEEE, 2016, pp. 162–167.

[20] H.A. Rahmani, M. Aliannejadi, S. Ahmadian, M. Baratchi, M. Afsharchi, F. Crestani, LGLMF: Local geographical based logistic matrix factorization model for POI recommendation, in: Asia Information Retrieval Symposium, Springer, 2019, pp. 66–78.

[21] X. Zhang, T. Wang, W. Luo, P. Huang, Multi-level fusion and attention-guided CNN for image dehazing, IEEE Trans. Circuits Syst. Video Technol. (2020).

[22] E.H. Houssein, B.E.-d. Helmy, D. Oliva, A.A. Elngar, H. Shaban, A novel black widow optimization algorithm for multilevel thresholding image segmentation, Expert Syst. Appl. 167 (2021) 114159.

[23] O. Ramos-Soto, E. Rodríguez-Esparza, S.E. Balderas-Mata, D. Oliva, A.E. Hassanien, R.K. Meleppat, R.J. Zawadzki, An efficient retinal blood vessel segmentation in eye fundus images by using optimized top-hat and homomorphic filtering, Comput. Methods Programs Biomed. (2021) 105949.

[24] N. Ortega-Sánchez, D. Oliva, E. Cuevas, M. Pérez-Cisneros, A.A. Juan, An evolutionary approach to improve the halftoning process, Mathematics 8 (9) (2020) 1636.

[25] D. Yousri, M. Abd Elaziz, L. Abualigah, D. Oliva, M.A. Al-Qaness, A.A. Ewees, COVID-19 X-ray images classification based on enhanced fractional-order cuckoo search optimizer using heavy-tailed distributions, Appl. Soft Comput. 101 (2021) 107052.

[26] M. Khodayar, M.E. Khodayar, S.M.J. Jalali, Deep learning for pattern recognition of photovoltaic energy generation, Electr. J. 34 (1) (2021) 106882.

[27] S.M.J. Jalali, P.M. Kebria, A. Khosravi, K. Saleh, D. Nahavandi, S. Nahavandi, Optimal autonomous driving through deep imitation learning and neuroevolution, in: 2019 IEEE International Conference on Systems, Man and Cybernetics, SMC, IEEE, 2019, pp. 1215–1220.

[28] X. Zhang, R. Jiang, T. Wang, J. Wang, Recursive neural network for video deblurring, IEEE Trans. Circuits Syst. Video Technol. (2020).

[29] M. Ahmadi, A. Sharifi, M. Jafarian Fard, N. Soleimani, Detection of brain lesion location in MRI images using convolutional neural network and robust PCA, Int. J. Neurosci. (2021) 1–12.

[30] M. Ahmadi, A. Sharifi, S. Hassantabar, S. Enayati, QAIS-DSNN: Tumor area segmentation of MRI image with optimized quantum matched-filter technique and deep spiking neural network, BioMed Res. Int. 2021 (2021).

[31] S. Dorosti, S.J. Ghoushchi, E. Sobhrakhshankhah, M. Ahmadi, A. Sharifi, Application of gene expression programming and sensitivity analyses in analyzing effective parameters in gastric cancer tumor size and location, Soft Comput. 24 (13) (2020) 9943–9964.

[32] E.E.-D. Hemdan, M.A. Shouman, M.E. Karar, Covidx-net: A framework of deep learning classifiers to diagnose COVID-19 in X-ray images, 2020, arXiv preprint arXiv:2003.11055.

[33] L. Wang, A. Wong, COVID-Net: A tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images, 2020, arXiv preprint arXiv:2003.09871.

[34] M. Loey, F. Smarandache, N.E.M. Khalifa, Within the lack of chest COVID-19 X-ray dataset: A novel detection model based on GAN and deep transfer learning, Symmetry 12 (4) (2020) 651.

[35] I.D. Apostolopoulos, T.A. Mpesiana, COVID-19: Automatic detection from X-ray images utilizing transfer learning with convolutional neural networks, Phys. Eng. Sci. Med. (2020) 1.

[36] A. Narin, C. Kaya, Z. Pamuk, Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks, 2020, arXiv preprint arXiv:2003.10849.
[37] P.K. Sethy, S.K. Behera, Detection of coronavirus disease (COVID-19) based on deep features, Preprints 2020030300, 2020, 2020.
[38] T. Ozturk, M. Talo, E.A. Yildirim, U.B. Baloglu, O. Yildirim, U.R. Acharya, Automated detection of COVID-19 cases using deep neural networks with X-ray images, Comput. Biol. Med. (2020) 103792.
[39] S. Hassantabar, M. Ahmadi, A. Sharifi, Diagnosis and detection of infected tissue of COVID-19 patients based on lung X-ray image using convolutional neural network approaches, Chaos Solitons Fractals 140 (2020) 110170.
[40] T. Goel, R. Murugan, S. Mirjalili, D.K. Chakrabartty, OptCoNet: An optimized convolutional neural network for an automatic diagnosis of COVID-19, Appl. Intell. (2020) 1–16.
[41] S.M.J. Jalali, S. Ahmadian, A. Khosravi, S. Mirjalili, M.R. Mahmoudi, S. Nahavandi, Neuroevolution-based autonomous robot navigation: A comparative study, Cogn. Syst. Res. 62 (2020) 35–43.
[42] S.M.J. Jalali, R. Hedjam, A. Khosravi, A.A. Heidari, S. Mirjalili, S. Nahavandi, Autonomous robot navigation using moth-flame-based neuroevolution, in: Evolutionary Machine Learning Techniques, Springer, 2020, pp. 67–83.
[43] S.M.J. Jalali, A. Khosravi, R. Alizadehsani, S.M. Salaken, P.M. Kebria, R. Puri, S. Nahavandi, Parsimonious evolutionary-based model development for detecting artery disease, in: 2019 IEEE International Conference on Industrial Technology, ICIT, IEEE, 2019, pp. 800–805.
[44] S.J. Mousavirad, G. Schaefer, S.M.J. Jalali, I. Korovin, A benchmark of recent population-based metaheuristic algorithms for multi-layer neural network training, in: Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion, 2020, pp. 1402–1408.
[45] D. Oliva, E. Rodriguez-Esparza, M.S. Martins, M. Abd Elaziz, S. Hinojosa, A.A. Ewees, S. Lu, Balancing the influence of evolutionary operators for global optimization, in: 2020 IEEE Congress on Evolutionary Computation, CEC, IEEE, 2020, pp. 1–8.
[46] S. Ahmadian, A.R. Khanteymoori, Training back propagation neural networks using asexual reproduction optimization, in: 2015 7th Conference on Information and Knowledge Technology, IKT, IEEE, 2015, pp. 1–6.
[47] S.J. Mousavirad, S.M.J. Jalali, S. Ahmadian, A. Khosravi, G. Schaefer, S. Nahavandi, Neural network training using a biogeography-based learning strategy, in: International Conference on Neural Information Processing, Springer, Cham, 2020, pp. 147–155.
[48] S.M.J. Jalali, S. Ahmadian, P.M. Kebria, A. Khosravi, C.P. Lim, S. Nahavandi, Evolving artificial neural networks using butterfly optimization algorithm for data classification, in: International Conference on Neural Information Processing, Springer, Cham, 2019, pp. 596–607.
[49] D. Zhao, L. Liu, F. Yu, A.A. Heidari, M. Wang, G. Liang, K. Muhammad, H. Chen, Chaotic random spare ant colony optimization for multi-threshold image segmentation of 2D Kapur entropy, Knowl.-Based Syst. (2020) 106510.
[50] J. Hu, H. Chen, A.A. Heidari, M. Wang, X. Zhang, Y. Chen, Z. Pan, Orthogonal learning covariance matrix for defects of grey wolf optimizer: Insights, balance, diversity, and feature selection, Knowl.-Based Syst. 213 (2021) 106684.
[51] Y. Zhang, R. Liu, A.A. Heidari, X. Wang, Y. Chen, M. Wang, H. Chen, Towards augmented kernel extreme learning models for bankruptcy prediction: Algorithmic behavior and comprehensive analysis, Neurocomputing (2020).
[52] W. Shan, Z. Qiao, A.A. Heidari, H. Chen, H. Turabieh, Y. Teng, Double adaptive weights for stabilization of moth flame optimizer: Balance analysis, engineering cases, and medical diagnosis, Knowl.-Based Syst. 214 (2021) 106728.
[53] H. Chen, A.A. Heidari, H. Chen, M. Wang, Z. Pan, A.H. Gandomi, Multi-population differential evolution-assisted Harris hawks optimization: Framework and case studies, Future Gener. Comput. Syst. 111 (2020) 175–198.
[54] J. Tu, H. Chen, J. Liu, A.A. Heidari, X. Zhang, M. Wang, R. Ruby, Q.-V. Pham, Evolutionary biogeography-based whale optimization methods with communication structure: Towards measuring the balance, Knowl.-Based Syst. 212 (2021) 106642.
[55] S.M.J. Jalali, S. Ahmadian, M. Khodayar, A. Khosravi, V. Ghasemi, M. Shafie-khah, S. Nahavandi, J.a. PS Catalão, Towards novel deep neuroevolution models: Chaotic levy grasshopper optimization for short-term wind speed forecasting, Eng. Comput. (2021) 1–25.
[56] S.M.J. Jalali, S. Ahmadian, A. Khosravi, M. Shafie-khah, S. Nahavandi, J.a. PS Catalão, A novel evolutionary-based deep convolutional neural network model for intelligent load forecasting, IEEE Trans. Ind. Inf. (2021) 1–8.
[57] A.W. Mohamed, A.A. Hadi, A.K. Mohamed, Gaining-sharing knowledge based algorithm for solving optimization problems: A novel nature-inspired algorithm, Int. J. Mach. Learn. Cybern. (2019) 1–29.
[58] M. Tubishat, N. Idris, L. Shuib, M.A. Abushariah, S. Mirjalili, Improved salp swarm algorithm based on opposition based learning and novel local search algorithm for feature selection, Expert Syst. Appl. 145 (2020) 113122.
[59] S. Gupta, K. Deep, A hybrid self-adaptive sine cosine algorithm with opposition based learning, Expert Syst. Appl. 119 (2019) 210–230.
[60] A.H. Niknamfar, S.T.A. Niaki, S.A.A. Niaki, Opposition-based learning for competitive hub location: A bi-objective biogeography-based optimization algorithm, Knowl.-Based Syst. 128 (2017) 1–19.
[61] Y. Feng, G.-G. Wang, J. Dong, L. Wang, Opposition-based learning monarch butterfly optimization with Gaussian perturbation for large-scale 0-1 knapsack problem, Comput. Electr. Eng. 67 (2018) 454–468.
[62] M. Ali, M. Pant, Improving the performance of differential evolution algorithm using Cauchy mutation, Soft Comput. 15 (5) (2011) 991–1007.
[63] Q. Wu, R. Law, Cauchy mutation based on objective variable of Gaussian particle swarm optimization for parameters selection of SVM, Expert Syst. Appl. 38 (6) (2011) 6405–6411.
[64] Y. Xu, H. Chen, J. Luo, Q. Zhang, S. Jiao, X. Zhang, Enhanced moth-flame optimizer with mutation strategy for global optimization, Inform. Sci. 492 (2019) 181–203.
[65] A.M. Alqudah, S. Qazan, Augmented COVID-19 X-ray images dataset, Mendeley Data, v4, 2020, http://dx.doi.org/10.17632/2fxz4px6d8, 4.
[66] S. Saremi, S. Mirjalili, A. Lewis, Grasshopper optimisation algorithm: Theory and application, Adv. Eng. Softw. 105 (2017) 30–47.
[67] S. Li, H. Chen, M. Wang, A.A. Heidari, S. Mirjalili, Slime mould algorithm: A new method for stochastic optimization, Future Gener. Comput. Syst. 111 (2020) 300–323.
[68] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (2014) 46–61.
[69] D. Simon, Biogeography-based optimization, IEEE Trans. Evol. Comput. 12 (6) (2008) 702–713.
[70] W. Deng, H. Liu, J. Xu, H. Zhao, Y. Song, An improved quantum-inspired differential evolution algorithm for deep belief network, IEEE Trans. Instrum. Meas. (2020).
[71] T.Y. Tan, L. Zhang, C.P. Lim, Intelligent skin cancer diagnosis using improved particle swarm optimization and deep learning models, Appl. Soft Comput. 84 (2019) 105725.