Machine Learning with Applications 21 (2025) 100720

Contents lists available at ScienceDirect

Machine Learning with Applications

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

Modelling the effect of vaccination on the spread of COVID-19 via a novel evolutionary ensemble learning algorithm

Mohammad Hassan Tayarani Najaran

University of Hertfordshire, Hatfield, UK

ARTICLE INFO

Keywords:
COVID-19
Ensemble learning
Evolutionary algorithms
Epidemiology
Optimisation
Policy-making
Vaccination

ABSTRACT

The spread of the COVID-19 disease has caused a lot of problems for every country around the world. To curb the pandemic, governments have issued various policies, including vaccination. Depending on the percentage of the vaccinated population, the pandemic responds differently to the policies. This paper proposes a modelling algorithm that takes as input the percentage of the vaccinated population and the policies taken by governments and generates as output a prediction of the number of newly infected cases. Then, this model is used as the fitness function in an optimisation algorithm, which for a population with a certain percentage of vaccinated people, searches through the set of policies and finds the best set of policies that minimises the cost to society and the number of infected people. To build the model, an ensemble learning algorithm is proposed, which is a combination of different learning algorithms. In this algorithm, an evolutionary diversifier algorithm is proposed to generate the base learners. The algorithm chooses different subsets of features for each base learner to maximise diversity among them. Then, an evolutionary process is adopted to choose from the base learners a subset that optimises the prediction accuracy of the model. The proposed algorithms are tested on a well-known data set about government policies, the percentage of the population vaccinated, and the number of infected cases. Experimental studies suggest better performance for the proposed ensemble learning algorithm compared to existing ones. Multi-objective optimisation of the policies is also proposed and tested on the model, and the results are presented in this paper.

1. Introduction

The spread of the COVID-19 disease has caused great havoc around the world, and countries have adopted a variety of policies to curb the pandemic. The invention of vaccines for the disease has promised hope for governments that the pandemic may soon be over. However, because vaccinating the whole population is not possible for many countries, and because the vaccines are not fully effective, governments still need to take measures to curb the pandemic. This makes it crucial to develop methods that predict the evolution of the pandemic concerning the proportion of the population that has been vaccinated and find the optimal solutions for controlling it. To build such a system, this paper proposes an evolutionary optimisation algorithm that, for a given percentage of the vaccinated population, searches through the set of policies and finds the optimal policies that minimise the growth rate of the pandemic and the costs incurred by society. To develop such an optimisation algorithm, a modelling algorithm should be built that can predict the growth rate of the pandemic as a result of a set of government policies. To do so, this paper proposes an ensemble algorithm that builds a model of the pandemic that takes as input the policies taken by the governments, as well as the proportion of the population that is vaccinated and predicts the growth rate of the infected cases. The optimisation algorithm then uses this model as a fitness function to find the optimal set of government policies.

1.1. Previous work

Because of their promising performance, ensemble learning approaches (Jin & Branke, 2005; Wang et al., 2019) have attracted the attention of many researchers. Much research, analytically (Dietterich, 2000) and experimentally (Cheng et al., 2020; Pratama et al., 2018; Yang et al., 2020; Zhu et al., 2020) have shown that an ensemble of base learning algorithms can outperform the individual algorithms. It is known that an ensemble can improve the performance of the base

E-mail address: m.tayaraninajaran@herts.ac.uk.

https://doi.org/10.1016/j.mlwa.2025.100720
Received 23 April 2025; Received in revised form 9 July 2025; Accepted 30 July 2025
Available online 8 August 2025

learning algorithms that are diverse (Lim et al., 2017; Najaran, 2023b; Polikar, 2006; Yang et al., 2014). An ensemble of algorithms is diverse if the base learners misclassify different instances. Diversity measures the error correlation among the base learners, which stems from the idea that if the classifiers in the ensemble make the same prediction, there is no point in building the ensemble (Najaran, 2023a).

Due to the importance of diversity in ensemble learning, much research targets the problem of how to improve diversity among base learners. In Sun and Zhou (2018), it is argued that most of the previous research only considers behavioural diversity, i.e. the results of classification rather than the inherent difference between classifiers. The authors consider structural diversity in addition to behavioural diversity when designing ensemble algorithms. It is argued in Mao et al. (2019) that diversity and individual accuracy are two contradictory objectives; thus, the authors propose a framework in which both objectives are taken into account. The diversity in ensemble learning is considered in Hayashi and Iiduka (2018) as a quadratic programming problem that is solved via a minimisation scheme. In Liu and Chen (2019), a method is proposed to increase diversity among the base learners for image classification tasks. It is argued in Kolárik et al. (2021), that diversity is crucial when it comes to designing ensemble learning in the presence of concept drift. To manage this, the authors present an adaptive approach that creates a set of heterogeneous base learners. A multi-objective evolutionary algorithm and a Bayesian Automatic Relevance Determination algorithm are proposed in Chen and Yao (2006) to promote diversity in ensemble learning algorithms. In Yang (2011), it is argued that it is better to use an ensemble of many, rather than all, classifiers; thus, classifier diversity has been used as a measure to choose from a set of classifiers.

One important approach to improve ensemble learning is pruning, which is to select from the set of base learners the classifiers that contribute the most to the ensemble's performance. This process improves the performance of the ensemble algorithm in terms of classification accuracy and computational cost. The simplest way of pruning is to prune the ensemble by removing the low-performance base-learners (for example, the method presented in Zhang, Cao, and Li 2021). In Dai et al. (2017), it is argued that existing ensemble pruning approaches consider diversity and accuracy separately, so a method is developed that simultaneously uses the two objectives in the pruning process. An ordering-based pruning method is presented in Xia et al. (2018), which ranks all the base learners concerning a new measure that evaluates the performance of each base learner as well as their complementarity to the ensemble. Traditionally, pruning techniques use a given criterion over the set of nearest neighbours to the validation data to estimate the competence of the base learners. In Oliveira et al. (2017), a dynamic selection is presented that detects if a validation sample is located in an indecision region and prunes the ensemble based on decision boundaries crossing the region of competence of the test sample. A method is presented in Li and Wen (2018), which treats the prediction results as features and manages the pruning as a feature selection method. The authors use the correlation between the target labels and predictions and the redundancy between classifiers to select the base learners. Other examples of ensemble pruning include multi-objective genetic algorithm (Zhang, Chen et al., 2021) and dynamic programming ensemble pruning (Alzubi et al., 2020; Li et al., 2019). A review of the application of AI methods on COVID-19, problems can be found in Tayarani (2020).

In this paper, we employ machine learning algorithms to build a model of the response of the COVID-19 pandemic to vaccination and use this model as a surrogate to optimise the set of policies taken by governments to curb the effects of the pandemic. This paper proposes an ensemble algorithm to build a model of the pandemic. The proposed ensemble algorithm consists of several different learning algorithms. It is shown in the literature that to reach a good performance in an ensemble of algorithms, there should be diversity among base learners. One way of creating diversity among base learners is to use a different subset of features for each base learner. To improve diversity, in this paper, an evolutionary algorithm is adopted that searches through different subsets of the feature set and chooses the subsets for each base learner in a way that makes the set of base learners diverse. It is shown in the literature that in ensemble algorithms, some of the base learners may be redundant. To reduce the number of redundant base learners, an evolutionary pruning algorithm is employed in this paper to select from the set of base learners a subset that optimises the performance of the final ensemble algorithm in terms of performance.

Table 1
The list of policies taken by countries. The correlation between each policy and the growth rate. This is based on the data for all the countries from 1st January 2020 until 1st June 2020.

Policy	Correlation	Value
P_1: Grocery and pharmacy stores	−0.201	[−100,100]
P_2: Parks and outdoor spaces	−0.181	[−100,100]
P_3 Retail and recreation	−0.317	[−100,100]
P_4 Time spent at home	−0.361	[−10,60]
P_5 Testing policy	−0.235	{0, 1, 2, 3}
P_6 Contact tracing	−0.111	{0, 1, 2}
P_7 Government Stringency Index	−0.521	[0,100]
P_8 Debt and contract relief	−0.311	{0, 1, 2}
P_9 Income support	−0.345	{0, 1, 2}
P_{10} Restrictions on internal movement	−0.322	{0, 1, 2}
P_{11} International travel controls	−0.337	{0, 1, 2, 3, 4}
P_{12} Public information campaigns	−0.424	{0, 1, 2}
P_{13} Cancellation of public events	−0.381	{0, 1, 2}
P_{14} Restrictions on public gatherings	−0.393	{0, 1, 2, 3, 4}
P_{15} Public transport stations	−0.244	[−100,50]
P_{16} School closures	−0.382	{0, 1, 2, 3}
P_{17} Stay-at-home restrictions	−0.321	{0, 1, 2, 3}
P_{18} Public transport	−0.431	{0, 1, 2}
P_{19} Workplace visitors	−0.408	[−100,100]
P_{20} Workplaces closures	−0.383	{0, 1, 2, 3}
P_{21} Vaccination	−0.482	{0, 100}

The rest of this paper is as follows. Section 2 presents the way the data are prepared. The proposed algorithms are presented in Section 2. Experimental analysis is performed in Section 4, and finally, Section 6 concludes the paper.

2. Data acquisition and preparation

In this paper, we use the data presented in Max Roser and Hasell (2020) to build a model of the pandemic and test the proposed algorithms. The source includes data about 21 policies (including vaccination) and the number of infected cases each day. Table 1 presents the list of the policies taken by the governments for which the data are available in this source. The data include information about the policies taken by governments from the 1st of January 2020, until the time of the writing of this paper, for the 21 policies, thus the data are a $d \times 21$ matrix, where d is the number of days. We assume in this paper that the growth rate in the number of new cases is affected by the policies taken by the government, and aim to build a model that finds the relationship between the policies and the number of positive COVID-19 cases. Surely, there are many factors other than the policies taken by the governments that affect the growth rate, examples of which include population density, public transport quality, weather, cultural behaviour of society, etc. However, it is very hard to take these factors into account as acquiring this data is an arduous task, and this paper aims to study the policies that are normally taken by governments.

In this paper, we take the policies devised by the governments as input features and take the growth rate of the COVID-19 positive cases as the output of the system. In this paper, we define the growth rate as the number of new cases on each day, divided by the number for the previous day.

There is some preprocessing that should be performed on the data before they are ready to be fed to modelling algorithms. Fig. 1 shows

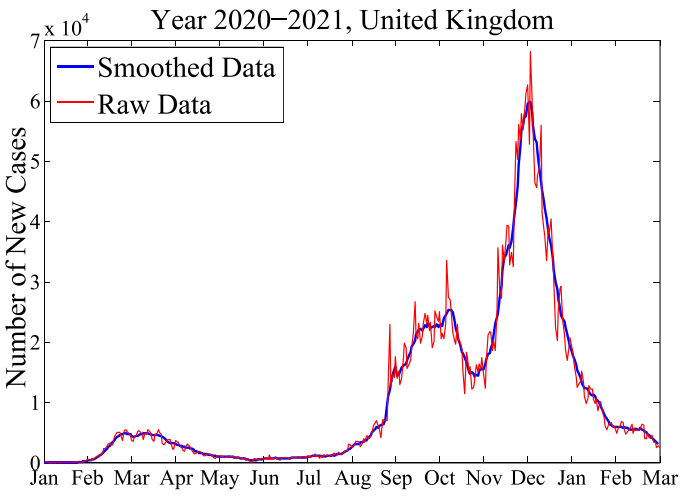

Fig. 1. The number of positive COVID-19 cases in the UK. The graphs show the reported cases and the smoothed graph. The graph is smoothed with a window of size 7 days.

the number of reported COVID-19-positive cases in the UK. As the graph suggests, for some reason, including the way data is collected, there are short-term fluctuations in the graph. These fluctuations are due to noise, rather than the true behaviour of the pandemic and should be managed as they can affect the performance of the modelling algorithms. To remove these fluctuations, the data are usually smoothed via the convolution operator, in which a window slides over the signal and, via averaging, removes the noise. The data in Fig. 1 are smoothed via a window of size 7, that is, the number of cases on each day is set to be the average number of cases from 3 days before to 3 days later. In this paper, we use a window size of 7 to smooth the graphs, as this is the value that is used in many analyses (see Max Roser and Hasell 2020). Also, our experiments suggest that this is the best value as the strongest correlation between the policies and the growth rate is observed when the window size is set to 7.

Another preprocessing required to be applied to the data is to find the time it takes for a policy to show its effect on the growth rate. This delay is due to several factors, including the incubation period of the virus. Thus, when preparing the data, the output of the policies for a particular day should be the growth rate of the pandemic on some days in the future. To study this, the correlation between the policies and the growth rate for a different number of days in the future is studied in this paper. Experiments suggest that the strongest correlation observed between the policies and the growth rate is around 14 days in the future. Table 1 presents the correlation between each of the policies and the growth rate when the convolution window size is 7 and the delay is set to 14.

3. Surrogate optimisation of government policies

From the beginning of the pandemic, governments around the world have taken different measures to control the spread of the virus. These measures usually harm the economy and people's mental health. The only way out of this conundrum is to reach community immunity (colloquially known as herd immunity) either via infection or vaccination. Naturally, after a certain proportion of society is infected by and recovered from a virus, community immunity is achieved. However, for a disease with a high death rate, this will result in the deaths of many people. Fortunately, for COVID-19, several vaccines have been developed, and immunity can be achieved by vaccination. This, however, cannot be performed in a short period due to many limitations, including the production capacity and the infrastructure required for vaccine injection. Until enough vaccine is produced to cover the entire society, the preventive measures still should be applied by the governments; the measures that induce huge costs on economies. In this sense, all governments around the world must find the optimal policies to minimise the number of cases and the economic impact on society at the same time. This paper aims to develop an intelligent algorithm to perform this task. The proposed algorithm first builds a model of the pandemic to predict its behaviour for a given set of policies when a particular proportion of society is vaccinated. Then an optimisation algorithm was adopted that searches among the set of different policies, and finds a set of policies that, for the given proportion of vaccination among people, minimise the growth rate and economic costs simultaneously.

In the proposed scheme, using machine learning techniques, a model of the pandemic's behaviour as a function of the policies taken by governments and the proportion of vaccination in society is generated. This model is then used as a surrogate to approximate the fitness function (growth rate) for a given set of policies. To build such a model, an ensemble learning algorithm is proposed in this paper. To reach a good performance via ensemble techniques, it is important to design and use a set of base learners with a high degree of diversity. As explained in Sun and Zhou (2018), two types of diversity should be taken into account when developing an ensemble, which are behavioural and inherent diversity. Behavioural diversity is achieved by manipulating the training data in a way that the base learners perform differently. Inherent diversity is achieved when the classifiers are different, where the diversity stems from the structural differences between the base learners.

There are many reasons why building a model of the pandemic is a hard task. First, the behaviour of the pandemic is very complicated and is affected by a large number of known, unknown, and sometimes unpredictable factors. Second, this paper aims to build a model of the pandemic based on a set of government policies and vaccination. This simplification makes the problem even harder, as many other factors are involved and affect the behaviour of the pandemic, including weather conditions, the density of population, age demography, vaccination against other viruses (flu vaccination is shown to reduce COVID-19 infection rate Conlon et al., 2021) type of COVID-19 vaccine used (vaccines produced by different companies have different performance), etc. These factors are extremely hard to take into account when building the model. Third, the data are not complete, as not all countries collect data with the same accuracy. Testing to discover the positive cases is an expensive task, and not every country has the required budget to collect sufficient information. Even if a sufficient budget is available, because many cases are asymptomatic and because the existing testing methods suffer from inaccuracy, there is always a difference between the true number of cases in a country and the reported number. Ensemble learning not only improves performance in terms of accuracy but also increases diversity, which in turn improves the robustness which is required in an uncertain environment.

Due to these reasons, the task is very hard for the existing algorithms, and there is a need to improve the performance of the learning algorithm. First, the modelling algorithm should be robust concerning the uncertainties explained before. Second, the performance in terms of accuracy should be improved in terms of accuracy. To achieve these, an ensemble learning algorithm is proposed in this paper, which improves the performance of the base learners by taking advantage of the wisdom of the crowd phenomenon.

This paper aims to develop the ensemble in a way that satisfies both types of diversity. To achieve inherent diversity, the ensemble learning consists of a set of different learning algorithms which include Feed-forward Neural Network (FNN), Function Fitting Neural Network (FFNN), Learning Vector Quantisation Neural Network (LVQ), Radial Basis Network (RBN), Cascade-Forward Neural Network (CFNN), Exact Radial Basis Network (RBE), Pattern Recognition Network (PRN), Generalised Regression Neural Network (GRNN), Probabilistic Neural Networks (PNN) and K-Nearest Neighbour (KNN). We performed some

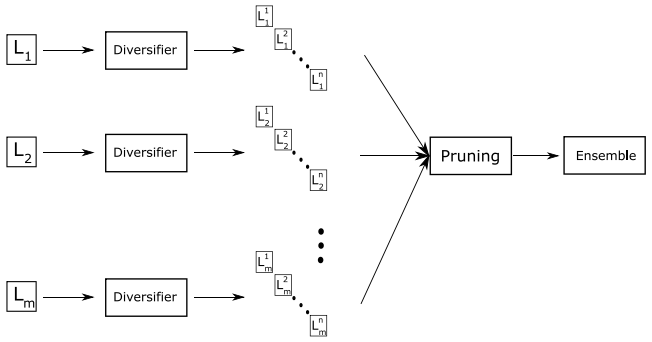

Fig. 2. The proposed diversification and pruning algorithm structure.

preliminary studies on a wide range of learning algorithms and finally found that the performance of these algorithms was adequate for our experiments. These algorithms were selected for three key reasons. First, they are among the most well-known and widely applied neural architectures in the machine learning and pattern recognition literature, making them strong and credible candidates for ensemble modelling. Second, all ten models are fully supported by MATLAB's Neural Network Toolbox, which allows for reliable implementation and consistency in training, evaluation, and comparison. Third, and most importantly, these learners represent a diverse set of modelling approaches, from distance-based methods and probabilistic inference to basis function expansion and supervised vector quantisation. Such diversity is vital for ensemble learning, as it increases the chances that the base models will capture different aspects of the data, thereby improving the overall generalisation capability of the ensemble. The structural differences between these classifiers offer inherent diversity. To achieve behavioural diversity, the manipulation of training data is adopted in this paper by training each base learner with a subset of features. It is important to arrange these subsets in a way that the base learners reach maximum performance and diversity at the same time. Finding the subsets that optimise these criteria is an optimisation task that is performed in this paper via an evolutionary algorithm. The resulting ensemble learning consists of a large number of base learners. Pruning (Dai et al., 2017) is adopted in this paper to reduce the number of base-learners (to reduce the computational cost) while retaining the performance of the learning algorithms. Thus, the pruning process is a multi-objective optimisation problem.

3.1. The ensemble diversifier algorithm

Fig. 2 shows the structure of the proposed diversification and pruning algorithms. The proposed method consists of m learning algorithms denoted by L_k, $k = 1 \dots m$. In the case of this paper, $m = 10$ and the list of learning algorithms include L_1 =FNN, L_2 =FFNN, L_3 =LVQ, L_4 =RBN, L_5 =CFNN, L_6 =RBE, L_7 =PRN, L_8 =GRNN, L_9 =PNN and L_{10} =KNN. The diversification stage gets as input one of the learning algorithms, L_k ($k = 1 \dots m$), and builds n base-learners denoted by $L_k^1 \dots L_k^n$ that are diverse. This process builds from the list of m learning algorithms, $m \times n$ base-learners that are diverse. Because the number of base learners is large, a pruning process is then adopted at the next stage that selects from the set of $m \times n$ base learners a subset that constructs the final ensemble algorithm. The output of the ensemble algorithm is the weighted sum of the base learners.

Algorithm 1 presents the evolutionary algorithm that is used for the diversification processes.

As presented in Fig. 2, the first step in the proposed algorithm is the diversifier. Improving the diversity among the base learners is performed in this module by assigning a subset of the features to each of the base learners in a way that maximises the diversity among the base learners and maximises their accuracy. This module gets as input the training and validation data and a classifier L_k and builds n subsets of the feature set. Thus, the optimisation problem is to find n different subsets of the feature sets in a way that the diversity and performance of the base learners using L_k are optimised. Because this is a combinatorial optimisation problem, we believe that Quantum Evolutionary Algorithm (QEA) is the most suitable algorithm to solve the problem. This is because QEAs are, in nature, mainly suitable for binary-coded combinatorial optimisation problems (Tayarani-N & Akbarzadeh-T, 2014).

Quantum Evolutionary Algorithm uses the principle of quantum computation and its superposition of states. QEA simulates certain principles of quantum mechanics, such as Q-bits, superposition, and rotation gates, within a classical computing environment. These quantum-inspired mechanisms allow the algorithm to maintain a probabilistic representation of individuals and enhance population diversity, thereby improving search effectiveness. This approach does not require actual quantum computation and can be efficiently implemented on conventional hardware. The unit of information in this algorithm is called a q-bit, which represents the probability of a bit being 0 or 1. In this sense, the solutions in this algorithm are strings of probability distribution functions (q-bits) that can represent binary strings. A string d q-bits is represented as,

$$q = \begin{bmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_j & \dots & \alpha_d \\ \beta_1 & \beta_2 & \dots & \beta_j & \dots & \beta_d \end{bmatrix}, \tag{1}$$

where $|\alpha_j|^2 + |\beta_j|^2 = 1$, $|\alpha_j|^2$ is the probability of the jth q-bit being zero and $|\beta_j|^2$ is the probability of it being one. Such representation allows the quantum individuals (which are each a string of q-bits) to represent the entire search space simultaneously. In QEA, the evolutionary process is performed via the "update" operator which changes a q-bit to represent zero or one with higher probability. The update operator applies on a q-bit as,

$$\begin{bmatrix} \alpha_j^{\tau+1} \\ \beta_j^{\tau+1} \end{bmatrix} = \begin{bmatrix} \cos(\Delta\theta) & -\sin(\Delta\theta) \\ \sin(\Delta\theta) & \cos(\Delta\theta) \end{bmatrix} \begin{bmatrix} \alpha_j^{\tau} \\ \beta_j^{\tau} \end{bmatrix}, \tag{2}$$

where $\Delta\theta$ is the rotation angle and controls the convergence speed of the algorithm, and τ is the iteration in the evolutionary process. For a more detailed description of the algorithm, please see Tayarani-N and Akbarzadeh-T (2014).

Although QEAs were originally designed for binary-coded combinatorial problems such as the knapsack problem, their underlying mechanisms make them particularly well-suited for high-dimensional tasks like feature selection and base learner pruning. In such tasks, the search space is typically vast and highly discrete, often with a large number of local optima. QEA's use of Q-bits allows each candidate solution to maintain a probabilistic representation over multiple states simultaneously, effectively encoding uncertainty and enabling parallel exploration of multiple regions in the search space. The use of rotation gates as update mechanisms provides a fine-grained, adaptive way to adjust probabilities based on fitness feedback, guiding the population towards promising areas without committing prematurely to suboptimal solutions. This balance of exploration and exploitation helps avoid premature convergence, a common challenge in high-dimensional spaces, and enables more efficient navigation towards globally competitive solutions. Compared to classical evolutionary algorithms, which rely on fixed mutation or crossover operators, QEA's probabilistic representation and adaptive update scheme allow for more flexible and diverse sampling of candidate solutions, which is particularly advantageous for selecting informative subsets of features or pruning ensembles of base learners where the interactions between variables are complex and nonlinear.

A description of the evolutionary algorithm used for diversification purposes is as follows. The algorithm gets as input a learning algorithm L_k and generates n base-learners that are diverse. The base learners differ in the subset of features they use in the learning process. The solutions x^i (for $i = 1 \dots p$, where p is the number of solutions in

Algorithm 1: The proposed diversifier QEA algorithm.

1 $\tau = 0$;
2 gets as input the learning algorithm L_k;
3 initialize the population Q^0 ;
4 observe Q^0 to generate X^0;
5 evaluate X^0;
6 store X^0 into B^0;
7 **while** *not termination condition* **do**
8 $\tau = \tau + 1$;
9 observe Q^τ to generate X^τ;
10 evaluate X^τ;
11 find the best neighbour of each q-individual and store in b^i if it is better than b^i ;
12 update Q^τ using Q-gate;
13 **for** *all q-individuals* $q^{i\tau}$ *in* Q^τ **do**
14 **if** $q^{i\tau}$ *is converged* **then**
15 $H \leftarrow \emptyset$;
16 $H \leftarrow H \cup \{q^{i\tau}\}$;
17 **for** *all q-individuals* $q^{j\tau}$ *neighbour to* $q^{i\tau}$ **do**
18 **if** $q^{j\tau}$ *is converged and is similar to* $q^{i\tau}$ **then**
19 $H \leftarrow H \cup \{q^{j\tau}\}$;
20 **end**
21 **end**
22 keep the best q-individual in H and reinitialize the rest;
23 **end**
24 **end**
25 **end**
26 RETURN the best-observed solution.;

the population) in this optimisation problem are each n subsets of the feature set represented by a $21\times n$ binary matrix where 21 is the number of features and n is the number of base-learners that use the learning algorithm L_k. In this representation, if $x^i_{lf} = 1$, it means that the fth feature is selected for the lth base-learner L^l_k and $x^i_{lf} = 0$ means it is not selected.

In step 3 of the algorithm, the population is initialised. The population here is a set of p q-individuals,

$$Q^\tau = \{q^{i\tau}\}, i = 1 \dots p \tag{3}$$

where τ is the iteration number of the evolutionary process and each q-individual, $q^{i\tau}$ is a $21\times n$ matrix of q-bits. The initialisation is performed by setting the q-bits to a value to represent zero and one with the same probability as

$$\begin{bmatrix} \alpha^{i0}_{lf} \\ \beta^{i0}_{lf} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}. \tag{4}$$

In step 4 of the algorithm, the q-individuals in the population are observed to produce binary matrices of solutions. The observation is performed by using the q-bit probability distributions to generate binary numbers. To generate a binary solution x, the observation operator is performed as,

$$x^\tau_{lf} = \begin{cases} 0 & \text{if} \quad \mathrm{R}(0,1) < |\alpha^\tau_{lf}|^2 \\ 1 & \text{otherwise,} \end{cases} \tag{5}$$

for $l = 1 \dots n$ and $f = 1 \dots 21$, and $\mathrm{R}(.,.)$ is a uniform random number generator.

In step 5 of the algorithm, the binary solutions x^i, for $i = 1 \dots p$, are evaluated, where p is the number of individuals in the population. The aim of algorithm 1 is to get as input a learning algorithm L_k and generate n base-learners L^l_k, for $l = 1 \dots n$ that are optimised to produce higher performance and diversity. Thus, we have a two-objective optimisation problem. These two objectives are measured by training the base-learners $L^1_k \dots L^n_k$ based on the subset of features suggested by x^i on the training data (L^l_k is trained based on the subset of features suggested by x^i_l) and testing them on the validation data denoted by V.

The first objective, the performance, is straightforward to measure. The performance of a base-learner L_k is measured as the mean squared error (MSE) between the output of the learning algorithm and the true value of the signal, as,

$$E_x(L^l_k) = \frac{1}{|V|} \sum_{\forall y_u \in V} (L^l_k(y_u, x) - r_u)^2, \tag{6}$$

where $|V|$ is the validation data, r_u is the growth rate for the y_u data record in V and $L^l_k(y_u, x)$ is the estimated growth rate for the data record y_u via the learning algorithm. The notation $L^l_k(y_u, x)$ means that the base learner is trained via the feature set that is suggested by the binary solution x.

The second objective is diversity among the base learners. The diversity between two base learners is measured as the MSE between the outputs of the two base learners.

$$D_x(L^l_k, L^h_k) = \frac{1}{|V|} \sum_{\forall y_u \in V} (L^l_k(y_u, x) - L^h_k(y_u, x))^2, \tag{7}$$

where $L^l_k(y_u, x)$ and $L^h_k(y_u, x)$ are the outputs of the base learners L^l_k and L^h_l for the data record y_u respectively. The error ($E_x(L_k)$) of the individual base learners should be minimised, and the diversity among the base learners should be maximised. The first objective is defined as the sum of the error of the individual base-learners as,

$$\mathcal{E}_x(L_k) = \sum_{l=1}^{n} E_x(L^l_k), \tag{8}$$

and diversity of the ensemble is defined as the average pairwise diversity among the base-learners $L^1_k \dots L^n_k$,

$$\mathcal{D}_x(L_k) = \frac{1}{n(n-1)} \sum_{l=1}^{n} \sum_{h=l+1}^{n} D_x(L^l_k, L^h_k). \tag{9}$$

Like many multi-objective optimisation approaches, the fitness of a solution in the proposed algorithm is measured with respect to the other solutions in the population. The fitness of a solution in the population, $\mathcal{F}(x)$ is measured as,

$$\mathcal{F}(x^i) = \sum_{j=1}^{p} [\mathcal{E}_{x^i}(L_k) < \mathcal{E}_{x^j}(L_k)] + \sum_{j=1}^{p} [\mathcal{D}_{x^i}(L_k) > \mathcal{D}_{x^j}(L_k)] \tag{10}$$

where [*statement*] returns 1 if the *statement* is true and returns 0 if it is false. This function calculates the fitness of x^i by adding the number of solutions in the population that are outperformed by the solution x^i in terms of the error (Eq. (8)) and the number of solutions that are outperformed in terms of the diversity (Eq. (9)).

The set B^τ in QEA contains the best-observed solutions by each quantum individual until the iteration τ. In step 7 of the algorithm, the set of best-observed solutions is updated. The **while** loop in step 7 of the algorithm iterates until the termination condition is met. The termination condition in this paper is set as the maximum number of iterations.

The population in the proposed algorithm is structured as presented in Fig. 3. In this structure, each q-individual is connected to and interacts with four different q-individuals. This architecture provides better diversity among the solutions and offers better performance (for more details, please see Tayarani-N and Akbarzadeh-T 2014). In step 11, the best neighbour of each individual is found (in terms of the fitness $\mathcal{F}(x)$ in Eq. (10)). For each q-individual, if the fitness of any of its neighbours is better than the fitness of the q-individual's best-observed

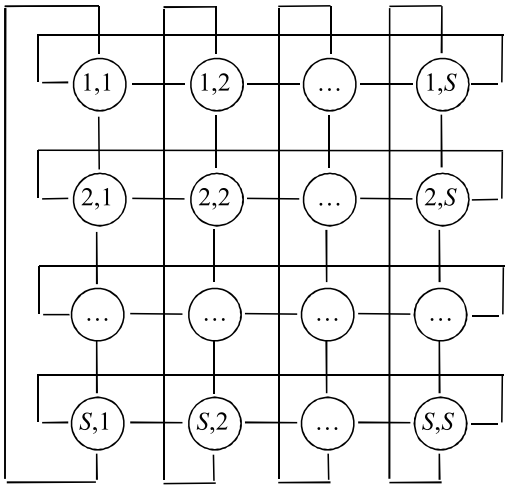

Fig. 3. The structure of the population in the proposed algorithm.

Algorithm 2: The proposed pruning QEA algorithm.

1 $\tau = 0$;
2 gets as input the set of base-learners L_k^l, $l = 1 \dots n$ and $k = 1 \dots m$;
3 initialize the population Q^0 based on Eq. (16);
4 observe Q^0 to generate Z^0;
5 evaluate Z^0;
6 store Z^0 into B^0;
7 **while** *not termination condition* **do**
8 $\tau = \tau + 1$;
9 observe Q^τ to generate Z^τ;
10 evaluate Z^τ;
11 find the best neighbour of each q-individual and store in b^i if it is better than b^i ;
12 update Q^τ using Q-gate;
13 **end**
14 RETURN the best-observed solution.;

solution ($\mathcal{F}(x) > \mathcal{F}(b^i)$), then the best neighbour is stored as b^i. Here, b^i contains the best solution that has been found by the q^i individual and its neighbours.

In step 13, all the q-individuals are checked to find the converged ones. A q-individual q^i is converged if,

$$\frac{1}{21n}\sum_{l=1}^{n}\sum_{f=1}^{21}\left|1-2\left|\alpha_{lf}^{i}\right|^{2}\right|>\gamma, \tag{11}$$

where γ is a parameter that controls the convergence criteria. If a q-individual is converged, in step 17, all its neighbours are found, and the ones that are similar to the q-individual and are converged are selected for re-initialisation. Two q-individuals x^i and x^j are similar if,

$$\frac{1}{21n}\sum_{l=1}^{n}\sum_{f=1}^{21}\left|x_{lf}^{i}-x_{lf}^{j}\right|<\lambda, \tag{12}$$

where λ is a parameter. This process maintains diversity in the population by detecting similar individuals and re-initialising them.

Algorithm 1 gets as input a learning algorithm L_k and generates n base-learners that are behaviourally diverse (Sun & Zhou, 2018). The proposed ensemble learning algorithm also benefits from structural diversity by adopting different learning algorithms.

3.2. The ensemble pruning algorithm

After the diversifier (algorithm 1) is performed, several $m \times n$ base-learners are generated that are optimised to be diverse and have improved performance. Here, the diversifier is performed independently on each of the learning algorithms, so the combined behaviour of all the $m \times n$ base learners is not considered in the optimisation process. Also, an ensemble with $m \times n$ base learners can computationally be expensive to implement on many systems. To overcome these shortcomings, a pruning algorithm is adopted after the base-learners are generated to select from among these $m \times n$ base-learners a subset that maximises the performance of the resulting ensemble learning. Removing the redundant base learners can not only reduce the computational cost but also improve the performance of the final ensemble.

In this paper, the evolutionary algorithm presented in algorithm 2 is used for the pruning process. A description of using this algorithm for pruning is as follows. In step 3 of the algorithm, the population is initialised. The solutions z in the pruning process are each a $m \times n$ matrix of zeros and ones, where $z_{kl} = 1$ means that the base-learner L_k^l is selected to participate in the decision process and $z_{kl} = 0$ means it is removed from the ensemble. In evolutionary algorithms, the initialisation process is usually performed randomly in a way that the entire search space is uniformly sampled. However, if there exists some prior knowledge that tells where in the search space it is more likely to find better solutions, the initialisation can be performed more efficiently. Here, we know that the base learners who bring more diversity to the ensemble and the ones with higher performance are more likely to be selected by the optimisation process. Therefore, we propose a method for the initialisation that produces quantum individuals that, with higher probability, select base-learners with higher performance and contribute more to the diversity. To do so, first, the performance of each base-learner L_k^l and the diversity it brings to the ensemble are estimated.

To find the performance of a base-learner, it is first trained on the training data, and then its performance is estimated based on the validation data. The performance of a base-learner L_k^l is found as Eq. (8), and the diversity it brings to the ensemble is estimated as the sum of pairwise diversity between L_k^l and all other base-learners in the ensemble,

$$\mathbb{D}(L_k^l)=\sum_{g=1}^{m}\sum_{h=1}^{n}D(L_k^l,L_g^h). \tag{13}$$

After the performance and the diversity that the base-learner L_k^l brings to the ensemble are calculated, the quality of each base-learner is calculated as,

$$\mathcal{G}(L_k^l)=\sum_{g=1}^{m}\sum_{h=1}^{n}[E(L_k^l)<E(L_h^g)]+\sum_{g=1}^{m}\sum_{h=1}^{n}[\mathbb{D}(L_k^l)>\mathbb{D}(L_h^g)], \tag{14}$$

where $[statement]$ returns 1 if the *statement* is true and returns 0 if it is false. The base learners are then sorted based on their quality calculated via Eq. (14). The ranking of the base-learner L_k^l based on its quality $\mathcal{G}(.)$ determines the probability with which the base-learner contributes to the ensemble ($z_{kl} = 1$). The q-individuals in the evolutionary algorithm are initialised in a way to produce binary solutions that contain high-quality base learners with higher probability. The rank of a base learner is found as,

$$\mathcal{R}(L_k^l)=\sum_{g=1}^{m}\sum_{h=1}^{n}[\mathcal{G}(L_k^l)<\mathcal{G}(L_h^g)]. \tag{15}$$

The rank of the base-learners is between 0 and $m \times n - 1$, where the highest quality base-learner (the base-learner with the largest $G(.)$) has

the rank of zero and the worst one is ranked $m\times n-1$. The q-individuals are then initialised as follows.

$$\begin{bmatrix}\alpha_{kl}^{i0}\\ \beta_{kl}^{i0}\end{bmatrix}=\begin{bmatrix}\sqrt{\frac{\mathcal{R}(L_k^l)}{m\times n-1}}\\ \sqrt{\frac{m\times n-\mathcal{R}(L_k^l)-1}{m\times n-1}}\end{bmatrix}. \tag{16}$$

In step 5 of algorithm 2 the binary solutions z_{kl} are evaluated. The evaluation of the binary solution z_{kl} is performed by training the ensemble algorithm on the training data and testing the performance on the validation data. The output of the ensemble for the data record y_u is determined as the weighted sum of the selected base learners as,

$$L(y_u,z)=\sum_{l=1}^{n}\sum_{k=1}^{m}z_{kl}w_{kl}L_k^l(y_u), \tag{17}$$

where $L(y_u,z)$ is the output of the ensemble algorithm for the data record y_u, z (which is a matrix of zeros and ones) determines what base-learners participate in the output and w_{kl} is the weight of L_k^l (the learning parameters of the ensemble). To find the optimal values for the ensemble weights, w, a simple gradient descent is adopted in this paper. The cost function for the optimisation process is defined as the mean square error between the target and the predicted value by the ensemble algorithm as

$$C(L,z)=\frac{1}{|T|}\sum_{\forall y_u\in T}(L(y_u,z)-r_u)^2, \tag{18}$$

where T is the set of training data. The gradient of the cost function with respect to the weights is thus

$$\frac{\partial C}{\partial w}=\frac{1}{|T|}\frac{\partial\sum_{\forall y_u\in T}(L(y_u,z)-r_u)^2}{\partial w}= \tag{19}$$

$$\frac{1}{|T|}\frac{\partial\sum_{\forall y_u\in T}(L(y_u,z)-r_u)^2}{\partial L(y_u,z)}\frac{\partial L(y_u,z)}{\partial w}= \tag{20}$$

$$\frac{2}{T}\sum_{\forall y_u\in T}\left((L(y_u,z)-r_u)\sum_{l=1}^{n}\sum_{k=1}^{m}z_{kl}L_k^l(y_u)\right). \tag{21}$$

The weights in the gradient descent process are updated as,

$$\Delta w=\frac{2\eta}{T}\sum_{\forall y_u\in T}\left((L(y_u,z)-r_u)\sum_{l=1}^{n}\sum_{k=1}^{m}z_{kl}L_k^l(y_u)\right). \tag{22}$$

After the ensemble is trained on the training data (the weights w_{kl} are optimised), it is evaluated on the validation data. The fitness here is simply defined as the mean square error of the ensemble on the validation data as presented in Eq. (8). The evolutionary optimisation in algorithm 2 progresses to find the best subset of base-learners.

After the optimal subset of base-learners is found via algorithm 2 the gradient descent is used to train the ensemble algorithm on the training data.

3.3. Evolutionary optimisation of government policies

This paper aims to build a surrogate model of the pandemic, an ensemble algorithm that gets as input the percentage of the population that has been vaccinated and a set of government policies and predicts the growth rate of the number of COVID-19-positive cases. Then, based on this surrogate model, an evolutionary algorithm can be adopted that, for a particular percentage of vaccinated society, can search through the set of all policies and find a set of policies that minimise the economic cost and the growth rate of the number of COVID-19 positive cases simultaneously. The optimisation algorithm uses the surrogate model of the pandemic to find the fitness of the solutions. The solutions here are each a set of policies taken by the government, and the fitness is measured based on the economic cost and the number of COVID-19 positive cases. Such an optimisation algorithm should satisfy a set of requirements. First, we face a multi-objective optimisation, where there are two conflicting objectives: the economic cost of the policies and the growth rate of the pandemic. Therefore, a multi-optimisation algorithm should be devised to manage this problem.

Second, because a surrogate model is used, the fitness of solutions is measured with uncertainty, a type of uncertainty known as approximation uncertainty (Jin & Branke, 2005; Tayarani-N. et al., 2015). Managing uncertainty is crucial as it can mislead the search process. There has been a wide range of research on how to reduce uncertainty when evolutionary algorithms have been adopted (Rakshit et al., 2017). The main approach in this area is to use a variety of averaging approaches to reduce uncertainty via resampling the fitness function. In the case of approximation uncertainty, however, resampling cannot be adopted as the surrogate models are deterministic and generate the same results for a particular input. The uncertainty in this paper is managed by adopting an ensemble of learning algorithms. The weighted sum of several ensembles performs as a way of averaging over several samples.

In this section, we describe the evolutionary algorithm that is proposed to find the best set of policies given that a particular percentage of the population has been vaccinated. In this paper, algorithm 3 is adopted to optimise the policies taken by the government.

The proposed algorithm gets as input the percentage of the population that has been vaccinated and finds the best set of policies that minimises the growth rate of the pandemic and the cost induced to society. While the growth rate of the pandemic is straightforward to measure and is estimated via the proposed surrogate model, it is a hard task to define and calculate the cost of a set of policies to society. In this paper, two scenarios are considered for the optimisation task.

In the first scenario, the government has a good indication of the cost each policy incurs on society, which is calculated as,

$$C(x)=\sum_{i=1}^{|P|}g_i(x_i), \tag{23}$$

where $g_i(x_i)$ is the cost of implementing the policy P_i as suggested by x_i and P is the set of policies as presented in Table 1. For example, as shown in Table 1, x_1 can take a value in the range of [−100,100] and $g_1(x_1)$ is the cost of implementing the policy with the value presented by x_1. Because in this scenario $g_i(.)$ for $i=1\dots|P|$ are known, the cost incurred to society can be easily calculated.

While assuming that the governments can have a good estimation of the cost incurred on society due to each policy is reasonable, it is understandable that, in many cases, such information may not be available. Thus, we devise a second scenario in which there is no need to have the estimated cost of each policy. To make the task easier, in this scenario, the policymakers should rank the policies based on the cost they believe each policy will incur to society. In this scenario, there is no need to have an estimation of the costs, and the policymakers should only know which policy incurs more cost than another. The first scenario is the more straightforward optimisation problem to solve; however, it requires a good estimation of the costs each policy incurs. The second scenario is the less straightforward one, but it is easier for the policy-makers to use as there is no need to know the cost of each policy. In the second scenario, the cost of a set of policies is calculated as,

$$C(x)=\sum_{i=1}^{|P|}r_ix_i, \tag{24}$$

where r_i is the rank of the policy P_i and the value of x_i is normalised between [0,1] so the weight of all the policies is equal. The ranking is performed from the least to most significant policies; that is, the higher the importance of the policy, the larger its rank. This way, if a high importance policy (with a large rank) is imposed (has a value close to 1), the cost function will be affected more strongly.

A description of the algorithm 3 is as follows. In step 4, the population is initialised. In this algorithm, the solutions are each a vector

Table 2
The performance of different learning algorithms in terms of mean square error. This is for when single learning algorithms are used versus when the proposed diversification algorithm (algorithm 1) is used to build 10 base learners. The data are averaged over 30 independent runs.

	USA		Germany		UK		Italy		France	
	Single	Diversified	Single	Diversified	Single	Diversified	Single	Diversified	Single	Diversified
RBN	4.785e−03	8.242e−04	3.327e−03	5.200e−04	2.486e−03	5.127e−04	**1.984e−03**	**3.325e−04**	2.229e−03	3.859e−04
LVQ	5.594e−03	1.033e−03	3.800e−03	5.972e−04	2.946e−03	5.806e−04	2.183e−03	3.564e−04	2.646e−03	4.751e−04
PNN	5.946e−03	1.014e−03	3.819e−03	6.411e−04	3.080e−03	5.327e−04	2.207e−03	3.944e−04	2.545e−03	4.284e−04
RBE	5.959e−03	1.128e−03	3.844e−03	6.635e−04	3.055e−03	6.367e−04	2.201e−03	4.266e−04	2.676e−03	4.550e−04
CFNN	4.453e−02	6.587e−03	1.669e−03	2.868e−04	3.395e−03	6.909e−04	2.460e−02	3.776e−03	1.320e−03	2.436e−04
PRN	4.246e−02	7.099e−03	4.389e−02	6.638e−03	5.586e−02	1.022e−02	4.821e−02	8.795e−03	4.594e−02	7.890e−03
FFNN	1.084e−02	1.798e−03	**1.464e−03**	**2.525e−04**	**1.376e−03**	3.328e−04	5.581e−02	8.819e−03	1.656e−03	3.328e−04
GRNN	**1.646e−03**	**3.108e−04**	1.550e−03	2.774e−04	1.625e−03	3.387e−04	2.537e−03	4.169e−04	**1.224e−03**	**2.118e−04**
KNN	1.710e−03	3.874e−04	1.618e−03	2.892e−04	1.721e−03	**3.314e−04**	2.640e−03	4.603e−04	1.274e−03	2.336e−04
FNN	4.033e−02	6.708e−03	1.865e−03	3.595e−04	4.336e−03	8.069e−04	3.818e−03	7.149e−04	2.652e−03	4.556e−04
Best	1.646e−03	3.108e−04	1.464e−03	2.525e−04	1.376e−03	3.314e−04	1.984e−03	3.325e−04	1.224e−03	2.118e−04
Ensemble	4.822e−04	8.906e−05	4.216e−04	2.135e−04	4.344e−04	2.099e−04	9.859e−04	2.994e−04	3.332e−04	4.005e−04

Algorithm 3: The policy optimisation algorithm.

1 gets as input the percentage of the population that has been vaccinated;
2 $\tau = 0$;
3 set the parameters c_1, c_2 and w;
4 initialize the population Y and velocity V;
5 **while** *not termination condition* **do**
6 $\tau = \tau + 1$;
7 estimate the growth rate of all the solutions in Y^τ;
8 estimate the cost of the solutions in Y^τ to the society;
9 find the fitness of the individuals in Y^τ via SPEA2;
10 update velocity via:
$v_i = wv_i + c_1 R(0,1)(pbest_i - x_i) + c_2 R(0,1)(gbest - x_i)$;
11 Update the position of solutions via $x_i = x_i + v_i$ Update the set of non-dominated solutions;
12 **end**
13 RETURN the best-observed solution.;

of size 20 ($P_1 \dots P_{20}$), with the values in the ranges that are presented in Table 1. Note that the vaccination percentage here is not an optimisation policy. All the other policies are plans that can be taken by the governments, but the vaccination is a fixed percentage of people who have been vaccinated at a given time. In other words, the policies $P_1 \dots P_{20}$ are variables that can be adjusted by governments at any given time, but the vaccination percentage takes months to adjust.

In step 7, the growth rate of the pandemic due to each of the solutions in the population (set of policies) is estimated via the ensemble algorithm presented earlier in Fig. 2. In step 8, the cost of the policies suggested by the solutions in the population is calculated via Eq. (24).

4. Experimental results

In this section, we perform a set of experiments on the proposed algorithms. As per Fig. 2, the proposed algorithm in this paper has three main components, which are the base learners, the diversifier, and the pruning algorithms. To study the effect of each component, we perform an ablation scheme, in which the three components are removed one by one and their effect on the performance is studied. Table 2 presents the performance of the base learners and the diversified algorithms. This shows how adding the diversifier improves the performance of the base learners.

As presented in Fig. 2, the diversifier algorithm gets as input a learning algorithm L_k and generates n learning algorithms that differ in the subset of features they are trained with. We start by studying the performance of the proposed diversifier algorithm. Table 2 shows the performance of different learning algorithms in predicting the growth rate of the pandemic for different countries when a single learning algorithm is used versus when the proposed diversifier algorithm is used to generate n base-learners based on the learning algorithm (algorithm 1). The data in this table are averaged over 30 independent runs. The data for diversified base-learners is generated by using the diversifier algorithm to generate 10 different diverse base-learners ($n = 10$). The output of these 10 base learners is combined via a simple weighted averaging. As the data suggests in this table, the proposed diversifier algorithm can generate an ensemble algorithm that outperforms the single algorithms with a good margin. As presented in algorithm 1, we propose a QEA algorithm for diversification purposes.

The columns denoted by "Single" represent the performance of the learning algorithm without diversification, that is, the output of each of the individual learning algorithms. The columns denoted by "Diversified" represent the results when the algorithm 1 is used to generate 10 base-learners.

The best results in Table 2 are bolded, and the row "Best" presents the best results in each column. The last row of this table, denoted by 'Ensemble', shows the results of the weighted voting scheme, where the weights are optimised via a gradient descent algorithm. That is, for the column denoted by "Single", and for the last row denoted by 'Ensemble', an ensemble of the single learning algorithms, $L_1 \dots L_m$, without diversification is generated. The output of the ensemble is then set to be the weighted sum of the output of the learning algorithms as

$$L(y_u, z) = \sum_{k=1}^{m} w_k L_k(y_u), \tag{25}$$

where $L_k(y_u)$ is the output of the base-learner L_k for the data record y_u.

For the column denoted by "Diversified", the learning algorithms $L_1 \dots L_m$ are diversified using algorithm 1 and $m \times n$ base-learners are generated. For the last row of this column, which is denoted by "Ensemble", an ensemble of these $m \times n$ base learners is generated, the output of which is determined as,

$$L(y_u, z) = \sum_{k=1}^{m} \sum_{l=1}^{n} w_{kl} L_k^l(y_u). \tag{26}$$

The weights are determined via a gradient descent scheme.

A comparison between the performance for single and diversified learning algorithms in Table 2 suggests that an ensemble of the base learners generated via the proposed diversifier outperforms the single learning algorithms by a good margin. Also, the data suggest that an ensemble of diversified classifiers outperforms an ensemble of single learning algorithms.

Tables 3 and 4 present the detailed experiments between the single learning algorithms and when algorithm 1 is used to build an ensemble of algorithms. In this table, the mean, standard deviation, and the p-value of the Wilcoxon rank-sum test between the experiments are presented.

Table 3
The performance of different learning algorithms in terms of mean square error. This is for when single learning algorithms are used versus when the proposed diversification algorithm is used to build 10 base learners. The data are averaged over 30 independent runs.

	USA				
	Single		Diversified		*p*-value
	Mean	STD	Mean	STD	
RBN	4.785e−03	7.720e−03	8.242e−04	1.639e−04	4.662e−03
LVQ	5.594e−03	7.482e−03	1.033e−03	1.914e−04	4.193e−05
PNN	5.946e−03	6.959e−03	1.014e−03	1.697e−04	1.320e−05
RBE	5.959e−03	9.475e−03	1.128e−03	1.431e−04	8.715e−05
CFNN	4.453e−02	3.916e−02	6.587e−03	1.253e−03	2.817e−06
PRN	4.246e−02	6.371e−02	7.099e−03	8.985e−04	9.644e−04
FFNN	1.084e−02	1.748e−02	1.798e−03	3.332e−04	4.836e−03
GRNN	1.646e−03	2.454e−03	3.108e−04	4.410e−05	1.600e−01
KNN	1.710e−03	2.586e−03	3.874e−04	4.206e−05	6.870e−06
FNN	4.033e−02	5.282e−02	6.708e−03	1.298e−03	1.326e−04
	Germany				
	Single		Diversified		*p*-value
	Mean	STD	Mean	STD	
RBN	3.327e−03	5.325e−03	5.200e−04	9.482e−05	1.010e−01
LVQ	3.800e−03	2.765e−03	5.972e−04	1.154e−04	1.038e−06
PNN	3.819e−03	3.558e−03	6.411e−04	1.050e−04	2.522e−06
RBE	3.844e−03	6.322e−03	6.635e−04	1.318e−04	4.836e−05
CFNN	1.669e−03	2.236e−03	2.868e−04	4.673e−05	1.270e−01
PRN	4.389e−02	5.308e−02	6.638e−03	9.486e−04	1.057e−03
FFNN	1.464e−03	1.935e−03	2.525e−04	3.224e−05	4.957e−04
GRNN	1.550e−03	1.856e−03	2.774e−04	5.074e−05	1.915e−02
KNN	1.618e−03	2.122e−03	2.892e−04	5.581e−05	1.312e−02
FNN	1.865e−03	1.512e−03	3.595e−04	7.027e−05	1.025e−06
	UK				
	Single		Diversified		*p*-value
	Mean	STD	Mean	STD	
RBN	2.486e−03	2.328e−03	5.127e−04	5.983e−05	1.788e−06
LVQ	2.946e−03	2.114e−03	5.806e−04	6.853e−05	1.308e−09
PNN	3.080e−03	4.703e−03	5.327e−04	1.038e−04	6.908e−03
RBE	3.055e−03	3.643e−03	6.367e−04	1.039e−04	2.080e−03
CFNN	3.395e−03	4.117e−03	6.909e−04	1.068e−04	1.687e−04
PRN	5.586e−02	5.174e−02	1.022e−02	1.683e−03	2.152e−06
FFNN	1.376e−03	1.214e−03	3.328e−04	6.624e−05	1.405e−06
GRNN	1.625e−03	2.636e−03	3.387e−04	6.141e−05	5.176e−02
KNN	1.721e−03	2.832e−03	3.314e−04	6.342e−05	6.412e−02
FNN	4.336e−03	5.445e−03	8.069e−04	1.258e−04	1.171e−02
	Italy				
	Single		Diversified		*p*-value
	Mean	STD	Mean	STD	
RBN	1.984e−03	2.627e−03	3.325e−04	4.365e−05	1.949e−02
LVQ	2.183e−03	2.894e−03	3.564e−04	5.146e−05	6.264e−05
PNN	2.207e−03	2.190e−03	3.944e−04	4.993e−05	1.000e−04
RBE	2.201e−03	3.524e−03	4.266e−04	4.875e−05	6.354e−03
CFNN	2.460e−02	1.771e−02	3.776e−03	6.672e−04	2.957e−07
PRN	4.821e−02	4.186e−02	8.795e−03	1.415e−03	8.169e−04
FFNN	5.581e−02	6.936e−02	8.819e−03	1.074e−03	2.007e−05
GRNN	2.537e−03	2.043e−03	4.169e−04	6.125e−05	5.451e−06
KNN	2.640e−03	1.952e−03	4.603e−04	8.262e−05	1.384e−09
FNN	3.818e−03	3.905e−03	7.149e−04	1.134e−04	4.167e−03
	France				
	Single		Diversified		*p*-value
	Mean	STD	Mean	STD	
RBN	2.229e−03	3.613e−03	3.859e−04	7.384e−05	3.122e−02
LVQ	2.646e−03	3.717e−03	4.751e−04	8.325e−05	1.025e−02
PNN	2.545e−03	3.407e−03	4.284e−04	7.751e−05	5.188e−04
RBE	2.676e−03	2.105e−03	4.550e−04	7.859e−05	3.748e−05
CFNN	1.320e−03	1.228e−03	2.436e−04	4.494e−05	1.085e−04
PRN	4.594e−02	4.692e−02	7.890e−03	1.275e−03	1.746e−04
FFNN	1.656e−03	1.837e−03	3.328e−04	5.370e−05	1.193e−05
GRNN	1.224e−03	9.874e−04	2.118e−04	2.138e−05	3.345e−05
KNN	1.274e−03	1.804e−03	2.336e−04	3.652e−05	1.064e−04
FNN	2.652e−03	4.069e−03	4.556e−04	8.447e−05	3.714e−03

Table 4
The performance of different learning algorithms in terms of mean square error. This is for when single learning algorithms are used versus when the proposed diversification algorithm is used to build 10 base-learners. The data are averaged over 30 independent runs.

	India				
	Single		Diversified		*p*-value
	Mean	STD	Mean	STD	
RBN	4.593e−03	5.464e−03	7.830e−04	8.626e−05	1.007e−07
LVQ	6.012e−03	9.481e−03	9.655e−04	1.927e−04	1.919e−02
PNN	5.582e−03	5.879e−03	1.025e−03	1.365e−04	2.347e−06
RBE	5.875e−03	6.753e−03	1.017e−03	1.319e−04	1.007e−04
CFNN	4.588e−02	7.632e−02	7.598e−03	8.069e−04	4.146e−02
PRN	4.447e−02	3.301e−02	6.494e−03	8.431e−04	3.446e−09
FFNN	1.054e−02	1.764e−02	1.815e−03	1.899e−04	2.989e−02
GRNN	1.361e−03	1.210e−03	2.820e−04	4.245e−05	2.731e−05
KNN	1.456e−03	1.990e−03	3.301e−04	5.814e−05	1.843e−04
FNN	4.369e−02	5.621e−02	6.711e−03	1.095e−03	4.822e−04
	Brazil				
	Single		Diversified		*p*-value
	Mean	STD	Mean	STD	
RBN	3.137e−03	4.314e−03	5.831e−04	6.355e−05	1.439e−06
LVQ	4.021e−03	4.266e−03	6.901e−04	7.459e−05	7.197e−05
PNN	3.849e−03	5.082e−03	7.501e−04	1.333e−04	1.382e−04
RBE	4.237e−03	6.403e−03	8.417e−04	1.604e−04	1.823e−01
CFNN	1.554e−03	1.118e−03	2.778e−04	4.260e−05	4.100e−05
PRN	4.344e−02	3.405e−02	7.429e−03	8.239e−04	1.138e−05
FFNN	1.356e−03	2.270e−03	2.602e−04	4.751e−05	5.667e−04
GRNN	1.727e−03	2.334e−03	2.909e−04	3.893e−05	8.638e−05
KNN	1.400e−03	1.303e−03	2.737e−04	3.542e−05	8.491e−06
FNN	1.970e−03	2.174e−03	3.812e−04	6.657e−05	5.278e−05
	South Korea				
	Single		Diversified		*p*-value
	Mean	STD	Mean	STD	
RBN	2.626e−03	2.159e−03	4.750e−04	4.799e−05	2.809e−11
LVQ	3.172e−03	3.072e−03	5.308e−04	5.565e−05	2.104e−04
PNN	2.757e−03	2.641e−03	4.304e−04	7.179e−05	8.663e−07
RBE	2.986e−03	3.081e−03	5.593e−04	8.967e−05	8.359e−07
CFNN	3.187e−03	2.716e−03	5.468e−04	8.345e−05	2.354e−03
PRN	5.917e−02	6.201e−02	9.328e−03	1.613e−03	1.162e−03
FFNN	1.460e−03	1.200e−03	2.800e−04	4.781e−05	2.152e−05
GRNN	1.779e−03	2.819e−03	3.523e−04	6.276e−05	6.106e−04
KNN	1.626e−03	1.292e−03	2.925e−04	3.767e−05	1.904e−06
FNN	4.132e−03	6.735e−03	6.396e−04	1.082e−04	5.826e−03
	China				
	Single		Diversified		*p*-value
	Mean	STD	Mean	STD	
RBN	1.846e−03	2.029e−03	2.897e−04	4.510e−05	8.480e−06
LVQ	2.013e−03	1.505e−03	3.705e−04	5.174e−05	2.849e−07
PNN	2.364e−03	2.464e−03	4.271e−04	4.534e−05	4.147e−08
RBE	2.253e−03	3.236e−03	3.955e−04	7.041e−05	1.411e−03
CFNN	2.482e−02	3.709e−02	4.427e−03	5.922e−04	9.293e−04
PRN	4.481e−02	5.578e−02	7.041e−03	1.132e−03	7.155e−03
FFNN	5.974e−02	8.281e−02	1.026e−02	1.787e−03	4.845e−05
GRNN	2.582e−03	4.115e−03	4.058e−04	4.483e−05	2.563e−03
KNN	2.533e−03	1.912e−03	4.364e−04	4.922e−05	5.077e−05
FNN	3.816e−03	3.830e−03	6.956e−04	1.078e−04	1.413e−04
	Israel				
	Single		Diversified		*p*-value
	Mean	STD	Mean	STD	
RBN	2.166e−03	1.616e−03	4.135e−04	6.142e−05	3.515e−05
LVQ	2.307e−03	2.066e−03	3.982e−04	7.528e−05	1.748e−04
PNN	2.487e−03	3.533e−03	4.695e−04	8.446e−05	6.494e−04
RBE	2.453e−03	3.487e−03	4.170e−04	7.232e−05	5.828e−04
CFNN	1.339e−03	2.113e−03	2.585e−04	2.718e−05	2.804e−02
PRN	4.345e−02	5.573e−02	6.879e−03	7.380e−04	3.387e−03
FFNN	1.585e−03	1.221e−03	2.811e−04	3.060e−05	9.478e−08
GRNN	1.102e−03	1.789e−03	1.990e−04	3.579e−05	6.247e−03
KNN	1.385e−03	2.078e−03	2.735e−04	5.314e−05	2.500e−02
FNN	2.976e−03	2.934e−03	4.686e−04	7.889e−05	2.676e−03

Table 5
The experimental results on using different optimisation algorithms as diversifiers to build the base learners. The data are averaged over 30 runs.

		DPQEA	QEA	GA	PSO	DE	FES	MASW	LLSO	RCGA	p-value
	RBN	8.242e−04	1.182e−03	1.860e−03	2.498e−03	2.259e−03	2.649e−03	1.775e−03	3.167e−03	1.649e−03	1.7e−90
	LVQ	1.033e−03	1.399e−03	2.073e−03	2.746e−03	2.622e−03	2.809e−03	1.909e−03	3.772e−03	1.683e−03	1.7e−83
	PNN	1.014e−03	1.267e−03	2.009e−03	2.765e−03	2.786e−03	2.896e−03	1.963e−03	3.801e−03	1.728e−03	3.2e−78
	RBE	1.128e−03	1.303e−03	1.926e−03	2.673e−03	2.750e−03	3.116e−03	2.187e−03	3.768e−03	1.882e−03	1.1e−85
USA	CFNN	6.587e−03	9.301e−03	1.493e−02	1.927e−02	2.129e−02	2.130e−02	1.308e−02	2.874e−02	1.211e−02	9.5e−110
	PRN	7.099e−03	9.025e−03	1.375e−02	1.946e−02	1.761e−02	1.908e−02	1.429e−02	2.709e−02	1.092e−02	2.5e−103
	FFNN	1.798e−03	2.424e−03	3.581e−03	5.204e−03	4.960e−03	4.764e−03	3.830e−03	6.385e−03	2.887e−03	7.9e−97
	GRNN	3.108e−04	4.459e−04	7.120e−04	9.110e−04	8.473e−04	1.048e−03	6.576e−04	1.229e−03	5.904e−04	1.1e−107
	KNN	3.874e−04	4.323e−04	7.619e−04	9.884e−04	1.107e−03	1.053e−03	6.763e−04	1.359e−03	6.208e−04	3.0e−111
	FNN	6.708e−03	7.553e−03	1.372e−02	1.830e−02	1.640e−02	1.754e−02	1.296e−02	2.420e−02	1.120e−02	4.2e−92
	RBN	5.200e−04	6.104e−04	1.014e−03	1.421e−03	1.338e−03	1.718e−03	1.542e−03	1.947e−03	1.129e−03	9.9e−96
	LVQ	5.972e−04	7.360e−04	9.893e−04	1.682e−03	1.576e−03	2.161e−03	1.625e−03	1.950e−03	1.069e−03	5.2e−102
	PNN	6.411e−04	6.882e−04	1.021e−03	1.764e−03	1.502e−03	1.947e−03	1.569e−03	1.965e−03	1.281e−03	2.8e−96
	RBE	6.635e−04	7.222e−04	1.059e−03	1.576e−03	1.437e−03	2.237e−03	1.548e−03	1.965e−03	1.081e−03	5.7e−103
Germany	CFNN	2.868e−04	3.325e−04	4.688e−04	7.085e−04	6.997e−04	1.052e−03	7.266e−04	1.064e−03	5.561e−04	1.8e−116
	PRN	6.638e−03	7.950e−03	1.177e−02	2.034e−02	1.586e−02	2.145e−02	1.790e−02	2.176e−02	1.290e−02	1.7e−94
	FFNN	2.525e−04	2.766e−04	4.017e−04	6.171e−04	5.523e−04	7.862e−04	7.200e−04	7.974e−04	4.921e−04	2.3e−96
	GRNN	2.774e−04	3.160e−04	4.524e−04	6.930e−04	5.976e−04	8.137e−04	6.419e−04	8.826e−04	5.205e−04	7.5e−111
	KNN	2.892e−04	3.309e−04	4.465e−04	7.224e−04	6.078e−04	9.953e−04	7.727e−04	9.658e−04	4.888e−04	2.5e−100
	FNN	3.595e−04	3.914e−04	5.294e−04	8.594e−04	7.509e−04	9.651e−04	7.662e−04	1.077e−03	5.449e−04	1.1e−97
	RBN	5.127e−04	5.482e−04	8.778e−04	7.818e−04	1.080e−03	1.088e−03	9.539e−04	1.572e−03	8.718e−04	2.5e−82
	LVQ	5.806e−04	6.567e−04	9.771e−04	9.764e−04	1.398e−03	1.307e−03	1.047e−03	2.178e−03	1.017e−03	2.3e−105
	PNN	5.327e−04	6.138e−04	9.462e−04	1.033e−03	1.264e−03	1.288e−03	1.055e−03	1.958e−03	1.002e−03	3.5e−100
	RBE	6.367e−04	6.783e−04	1.057e−03	9.373e−04	1.244e−03	1.361e−03	9.627e−04	2.123e−03	1.009e−03	3.0e−93
UK	CFNN	6.909e−04	6.680e−04	1.072e−03	1.084e−03	1.465e−03	1.354e−03	1.218e−03	2.466e−03	1.116e−03	1.8e−96
	PRN	1.022e−02	1.073e−02	1.640e−02	1.624e−02	2.276e−02	2.355e−02	1.914e−02	3.666e−02	1.790e−02	1.8e−117
	FFNN	3.328e−04	3.532e−04	5.400e−04	5.485e−04	6.991e−04	7.508e−04	5.595e−04	1.084e−03	5.186e−04	7.2e−77
	GRNN	3.387e−04	3.819e−04	6.113e−04	6.175e−04	7.023e−04	7.419e−04	5.361e−04	1.117e−03	6.026e−04	6.4e−121
	KNN	3.314e−04	4.080e−04	6.846e−04	6.536e−04	8.933e−04	7.665e−04	5.699e−04	1.125e−03	6.417e−04	1.3e−90
	FNN	8.069e−04	8.606e−04	1.559e−03	1.522e−03	1.922e−03	1.983e−03	1.443e−03	2.607e−03	1.388e−03	4.5e−95
	RBN	3.325e−04	4.279e−04	5.420e−04	7.187e−04	1.128e−03	8.137e−04	6.036e−04	1.035e−03	5.513e−04	2.0e−100
	LVQ	3.564e−04	3.923e−04	5.400e−04	7.838e−04	1.292e−03	7.867e−04	6.743e−04	1.201e−03	6.330e−04	6.1e−131
	PNN	3.944e−04	3.929e−04	6.173e−04	7.648e−04	1.343e−03	9.037e−04	7.440e−04	1.010e−03	6.339e−04	5.1e−94
	RBE	4.266e−04	4.194e−04	5.672e−04	7.911e−04	1.244e−03	7.967e−04	7.738e−04	1.167e−03	5.926e−04	8.9e−96
Italy	CFNN	3.776e−03	5.029e−03	6.273e−03	7.842e−03	1.472e−02	9.643e−03	8.220e−03	1.171e−02	6.491e−03	5.4e−107
	PRN	8.795e−03	8.684e−03	1.238e−02	1.838e−02	2.447e−02	1.702e−02	1.453e−02	2.308e−02	1.269e−02	5.0e−100
	FFNN	8.819e−03	9.775e−03	1.552e−02	2.161e−02	3.310e−02	2.243e−02	1.692e−02	2.937e−02	1.639e−02	9.5e−109
	GRNN	4.169e−04	4.946e−04	7.220e−04	9.761e−04	1.374e−03	1.004e−03	7.683e−04	1.329e−03	6.489e−04	4.5e−92
	KNN	4.603e−04	5.031e−04	6.789e−04	9.229e−04	1.572e−03	1.015e−03	8.178e−04	1.379e−03	6.888e−04	8.5e−108
	FNN	7.149e−04	8.141e−04	1.110e−03	1.542e−03	2.233e−03	1.416e−03	1.310e−03	2.060e−03	1.115e−03	8.1e−86
	RBN	3.859e−04	4.832e−04	8.081e−04	1.035e−03	1.250e−03	1.093e−03	8.371e−04	1.220e−03	6.475e−04	2.1e−95
	LVQ	4.751e−04	5.244e−04	8.690e−04	1.131e−03	1.441e−03	1.360e−03	9.937e−04	1.465e−03	8.123e−04	2.0e−88
	PNN	4.284e−04	5.431e−04	8.214e−04	1.060e−03	1.333e−03	1.199e−03	1.102e−03	1.331e−03	7.695e−04	8.2e−90
	RBE	4.550e−04	5.665e−04	8.197e−04	1.215e−03	1.232e−03	1.358e−03	9.753e−04	1.453e−03	7.350e−04	6.2e−87
France	CFNN	2.436e−04	3.079e−04	5.093e−04	6.032e−04	7.327e−04	8.301e−04	5.318e−04	8.083e−04	4.347e−04	4.7e−104
	PRN	7.890e−03	1.009e−02	1.397e−02	1.988e−02	2.204e−02	2.022e−02	1.575e−02	2.440e−02	1.227e−02	1.8e−87
	FFNN	3.328e−04	3.542e−04	6.129e−04	7.776e−04	8.496e−04	8.606e−04	6.654e−04	9.319e−04	5.312e−04	5.9e−77
	GRNN	2.118e−04	2.621e−04	4.501e−04	5.643e−04	6.683e−04	6.768e−04	4.930e−04	6.730e−04	3.473e−04	3.0e−90
	KNN	2.336e−04	3.200e−04	4.940e−04	6.207e−04	7.201e−04	7.182e−04	5.187e−04	7.143e−04	4.133e−04	1.3e−90
	FNN	4.556e−04	6.029e−04	9.251e−04	1.137e−03	1.458e−03	1.387e−03	1.008e−03	1.579e−03	8.277e−04	4.3e−106
Rank		1	2	9	3	7	4	5	6	8	–

Due to the statistical nature of the optimisation algorithms, all results are averaged over 30 runs. To generate these data, the optimisation algorithms were used to build 10 base learners, the outputs of which are aggregated via a gradient descent algorithm. The last row of the table shows the Friedman rank of each of the algorithms based on the data in this table. As the data suggest, the best performance is achieved by the proposed DPQEA, followed by QEA. Here, QEA outperforms all the other algorithms, which we believe is because QEA is specifically efficient in solving combinatorial optimisation problems.

The proposed algorithm is compared with a group of existing algorithms, including Genetic Algorithms, PSO, Fast Evolutionary Strategy (FES) (Yao & Liu, 1997), Ma-ssw-chains (MASSW) (Molina et al., 2011), Differential Evolution (DE), Rotating Crossover Operator Differential Evolution (RCODE) (Deng et al., 2018), Level-Based Learning Swarm Optimiser (LLSO) (Yang et al., 2018), and Real-coded Compact Genetic Algorithms (RCGA) (Mininno et al., 2008). The population size for all algorithms for all the experiments is set to 10, and the termination condition is set for a maximum of 100 generations. Because tuning the parameters of these algorithms specifically for the problem in this paper is very time-consuming, the parameters of these algorithms are set to the parameters studied in Najaran and Tootounchi (2020).

Quantum-inspired Evolutionary Algorithms (QEAs) are a class of meta-heuristic optimisation techniques that draw inspiration from quantum computing principles such as quantum bits (Q-bits), superposition, and quantum gates. Unlike traditional evolutionary algorithms that use binary or real-valued encodings, QEAs use probabilistic representations where each Q-bit encodes a probability amplitude, allowing the algorithm to explore multiple states simultaneously. This facilitates better diversity and reduces premature convergence, especially in high-dimensional or combinatorial spaces. Genetic Algorithms (GAs) are classical evolutionary algorithms based on the process of natural selection, where candidate solutions evolve through operations such as selection, crossover, and mutation. Particle Swarm Optimisation (PSO) is inspired by the social behaviour of birds and fish, where particles (solutions) move through the search space influenced by

Table 6
The experimental results on using different optimisation algorithms as diversifiers to build the base learners. The data are averaged over 30 runs.

		DPQEA	QEA	GA	PSO	DE	FES	MASW	LLSO	RCGA	p-value
India	RBN	7.830e−04	9.508e−04	1.441e−03	1.486e−03	1.946e−03	1.440e−03	1.664e−03	2.243e−03	1.118e−03	5.0e−89
	LVQ	9.655e−04	1.292e−03	1.788e−03	1.983e−03	2.642e−03	1.939e−03	2.213e−03	2.484e−03	1.536e−03	2.8e−76
	PNN	1.025e−03	1.196e−03	1.660e−03	1.857e−03	2.436e−03	1.863e−03	1.924e−03	2.336e−03	1.470e−03	1.1e−74
	RBE	1.017e−03	1.279e−03	1.783e−03	2.004e−03	2.735e−03	1.787e−03	2.290e−03	2.899e−03	1.524e−03	1.8e−87
	CFNN	7.598e−03	8.859e−03	1.341e−02	1.314e−02	1.828e−02	1.317e−02	1.624e−02	1.861e−02	1.036e−02	2.8e−73
	PRN	6.494e−03	8.665e−03	1.286e−02	1.348e−02	1.682e−02	1.432e−02	1.516e−02	1.992e−02	1.125e−02	4.6e−80
	FFNN	1.815e−03	2.245e−03	2.836e−03	3.038e−03	4.303e−03	3.031e−03	3.507e−03	5.007e−03	2.823e−03	2.8e−82
	GRNN	2.820e−04	3.488e−04	5.213e−04	5.633e−04	7.603e−04	5.558e−04	5.939e−04	7.753e−04	4.591e−04	6.6e−77
	KNN	3.301e−04	4.338e−04	5.815e−04	7.069e−04	8.742e−04	6.996e−04	6.812e−04	9.922e−04	5.213e−04	4.9e−77
	FNN	6.711e−03	8.680e−03	1.199e−02	1.347e−02	1.903e−02	1.401e−02	1.427e−02	1.987e−02	9.694e−03	9.7e−93
Brazil	RBN	5.831e−04	6.354e−04	9.626e−04	1.079e−03	1.249e−03	9.467e−04	8.693e−04	1.147e−03	8.753e−04	2.8e−67
	LVQ	6.901e−04	8.080e−04	1.204e−03	1.300e−03	1.578e−03	1.348e−03	1.275e−03	1.744e−03	1.078e−03	6.7e−86
	PNN	7.501e−04	7.773e−04	1.106e−03	1.397e−03	1.553e−03	1.125e−03	1.170e−03	1.413e−03	1.153e−03	1.8e−56
	RBE	8.417e−04	9.370e−04	1.381e−03	1.569e−03	1.605e−03	1.232e−03	1.204e−03	1.598e−03	1.353e−03	6.0e−41
	CFNN	2.778e−04	3.557e−04	4.895e−04	5.911e−04	6.151e−04	4.740e−04	5.187e−04	6.027e−04	4.400e−04	3.4e−76
	PRN	7.429e−03	9.132e−03	1.450e−02	1.483e−02	1.477e−02	1.261e−02	1.206e−02	1.626e−02	1.236e−02	2.6e−57
	FFNN	2.602e−04	2.866e−04	4.259e−04	5.011e−04	5.806e−04	4.176e−04	3.927e−04	6.012e−04	4.100e−04	1.9e−62
	GRNN	2.909e−04	3.827e−04	5.165e−04	6.352e−04	6.457e−04	4.878e−04	5.229e−04	7.625e−04	5.439e−04	2.5e−59
	KNN	2.737e−04	3.081e−04	4.650e−04	5.610e−04	6.357e−04	4.233e−04	4.455e−04	5.444e−04	4.249e−04	1.4e−63
	FNN	3.812e−04	4.262e−04	6.110e−04	7.630e−04	7.204e−04	5.821e−04	6.262e−04	8.402e−04	6.267e−04	1.8e−73
South Korea	RBN	4.750e−04	5.103e−04	9.052e−04	9.674e−04	1.132e−03	8.420e−04	8.831e−04	1.327e−03	7.439e−04	2.3e−85
	LVQ	5.308e−04	6.292e−04	1.098e−03	1.154e−03	1.369e−03	9.166e−04	1.126e−03	1.376e−03	9.069e−04	2.2e−70
	PNN	4.304e−04	5.352e−04	9.227e−04	1.061e−03	1.234e−03	9.008e−04	9.435e−04	1.293e−03	8.170e−04	1.3e−81
	RBE	5.593e−04	5.614e−04	1.007e−03	1.184e−03	1.312e−03	8.490e−04	9.967e−04	1.243e−03	8.057e−04	1.3e−82
	CFNN	5.468e−04	6.829e−04	1.040e−03	1.254e−03	1.405e−03	9.054e−04	1.079e−03	1.457e−03	9.196e−04	2.4e−67
	PRN	9.328e−03	1.040e−02	1.829e−02	1.949e−02	2.521e−02	1.674e−02	1.796e−02	2.536e−02	1.441e−02	2.5e−75
	FFNN	2.800e−04	3.192e−04	4.972e−04	5.706e−04	6.344e−04	5.204e−04	5.117e−04	7.491e−04	4.625e−04	5.5e−79
	GRNN	3.523e−04	3.722e−04	5.672e−04	6.779e−04	7.490e−04	5.660e−04	6.012e−04	8.060e−04	5.518e−04	2.5e−70
	KNN	2.925e−04	3.204e−04	5.437e−04	7.008e−04	6.906e−04	4.915e−04	6.381e−04	7.975e−04	5.237e−04	8.2e−86
	FNN	6.396e−04	7.338e−04	1.178e−03	1.519e−03	1.699e−03	1.196e−03	1.510e−03	1.971e−03	1.209e−03	3.4e−99
China	RBN	2.897e−04	4.548e−04	5.898e−04	6.138e−04	8.161e−04	5.151e−04	6.330e−04	8.655e−04	5.457e−04	2.1e−92
	LVQ	3.705e−04	4.275e−04	7.201e−04	7.605e−04	8.100e−04	6.463e−04	7.194e−04	9.810e−04	5.360e−04	1.1e−81
	PNN	4.271e−04	5.595e−04	7.215e−04	8.062e−04	9.464e−04	6.561e−04	8.750e−04	1.064e−03	7.275e−04	2.3e−68
	RBE	3.955e−04	4.806e−04	7.224e−04	8.247e−04	8.551e−04	7.041e−04	8.434e−04	1.069e−03	6.656e−04	4.9e−77
	CFNN	4.427e−03	5.457e−03	7.179e−03	8.806e−03	9.412e−03	7.467e−03	8.198e−03	1.057e−02	6.700e−03	3.0e−62
	PRN	7.041e−03	9.478e−03	1.536e−02	1.421e−02	1.881e−02	1.182e−02	1.457e−02	2.163e−02	1.271e−02	1.7e−92
	FFNN	1.026e−02	1.203e−02	1.979e−02	1.965e−02	2.117e−02	1.795e−02	2.021e−02	2.527e−02	1.805e−02	2.3e−85
	GRNN	4.058e−04	5.402e−04	7.888e−04	9.562e−04	9.575e−04	8.439e−04	9.342e−04	1.216e−03	7.894e−04	6.5e−82
	KNN	4.364e−04	5.951e−04	8.881e−04	9.309e−04	9.951e−04	6.889e−04	8.902e−04	1.222e−03	6.892e−04	4.0e−79
	FNN	6.956e−04	8.826e−04	1.352e−03	1.337e−03	1.559e−03	1.111e−03	1.378e−03	1.765e−03	1.140e−03	1.0e−74
Israel	RBN	4.135e−04	5.482e−04	6.387e−04	6.218e−04	9.504e−04	6.251e−04	9.007e−04	9.521e−04	5.973e−04	6.4e−59
	LVQ	3.982e−04	5.770e−04	6.893e−04	7.509e−04	9.838e−04	7.592e−04	8.472e−04	9.885e−04	6.211e−04	3.7e−72
	PNN	4.695e−04	6.081e−04	6.486e−04	8.247e−04	1.028e−03	6.607e−04	8.849e−04	1.115e−03	6.275e−04	1.9e−67
	RBE	4.170e−04	5.337e−04	6.231e−04	7.857e−04	9.593e−04	6.899e−04	9.918e−04	9.409e−04	5.622e−04	1.3e−72
	CFNN	2.585e−04	3.292e−04	3.974e−04	4.264e−04	5.388e−04	3.950e−04	5.480e−04	5.823e−04	3.452e−04	2.5e−63
	PRN	6.879e−03	8.788e−03	1.109e−02	1.179e−02	1.532e−02	1.185e−02	1.609e−02	1.564e−02	9.619e−03	4.2e−78
	FFNN	2.811e−04	4.250e−04	4.434e−04	5.022e−04	6.846e−04	5.190e−04	6.681e−04	7.094e−04	4.058e−04	8.6e−84
	GRNN	1.990e−04	2.994e−04	3.045e−04	3.554e−04	4.431e−04	3.748e−04	4.527e−04	5.297e−04	3.178e−04	3.0e−74
	KNN	2.735e−04	3.737e−04	3.640e−04	4.422e−04	5.541e−04	4.017e−04	6.135e−04	6.198e−04	3.629e−04	1.7e−57
	FNN	4.686e−04	7.280e−04	8.619e−04	9.848e−04	1.090e−03	9.345e−04	1.072e−03	1.350e−03	7.531e−04	2.6e−57
Rank		1	2	9	6	3	7	4	5	8	–

their own experience and the experience of their neighbours, effectively balancing exploration and exploitation. Differential Evolution (DE) is a population-based stochastic optimiser that perturbs solutions based on the scaled differences of randomly selected individuals, making it particularly effective in continuous optimisation problems. Fast Evolutionary Strategies (FES) are variants of evolutionary algorithms designed to converge rapidly, often through adaptive mechanisms and aggressive selection strategies. The Memetic Algorithm with Solis-Wets Local Search (MASW) enhances global search capability with local refinement, combining evolutionary principles with a powerful local optimiser for fine-tuning solutions. Estimation of Distribution Algorithms (EDAs) replaces traditional crossover and mutation with probabilistic model building, sampling new individuals based on learned statistical dependencies in promising solutions. Level-Based Learning Swarm Optimiser (LLSO) integrates learning from hierarchical levels of solution quality, guiding search dynamically through swarm-based cooperation. Lastly, the Real-Coded Compact Genetic Algorithm (RCGA) is a variant of GAs that operates on real-valued vectors using probabilistic models to simulate crossover and mutation, enabling efficient search in continuous spaces. By employing and comparing this diverse suite of evolutionary and swarm-based optimisers, the study aims to evaluate the robustness and performance of the proposed diversification and ensemble strategy across a broad optimisation landscape.

To compare the performance of the proposed optimisation algorithm with state-of-the-art optimisation algorithms, Tables 5 and 6 show the experiments for the different optimisation algorithms. The data in this table suggests that the proposed DPQEA outperforms the other optimisation algorithms. Therefore, we use this algorithm to diversify the learning algorithms and report the results in Table 2. The last column in these tables presents the p-value for the ANOVA test between the results.

The p-value column in Table 6 presents the results of an ANOVA (Analysis of Variance) statistical test, which is used to assess whether the differences in performance between the proposed algorithm (DPQEA)

Table 7
The performance of different optimisation algorithms for pruning the ensemble algorithm. The data are averaged over 30 runs. The numbers in the brackets show the Friedman rank of the algorithms.

	USA	Germany	UK	Italy	France
DPQEA (1)	2.36e−05	6.23e−05	6.75e−05	9.20e−05	1.19e−04
QEA (2)	3.31e−05	5.99e−05	7.46e−05	1.05e−04	1.35e−04
GA (8)	3.76e−05	7.21e−05	8.81e−05	1.30e−04	1.94e−04
PSO (4)	3.72e−05	8.72e−05	8.67e−05	1.21e−04	1.39e−04
DE (5)	3.11e−05	6.60e−05	7.14e−05	1.59e−04	1.96e−04
FES (6)	4.47e−05	9.34e−05	7.28e−05	1.06e−04	1.57e−04
MASW (3)	3.04e−05	6.40e−05	8.29e−05	1.18e−04	1.47e−04
LLSO (7)	3.94e−05	9.77e−05	1.07e−04	1.02e−04	1.51e−04
RCGA (9)	5.11e−05	1.15e−04	1.36e−04	1.41e−04	2.04e−04
no pruning	8.90e−05	2.13e−04	2.09e−04	2.99e−04	4.00e−04

and the other competing algorithms are statistically significant. Specifically, a low p-value (typically less than 0.05) indicates that the observed differences in performance are unlikely to have occurred by chance alone and that there is strong evidence of a real difference between the algorithms. In this table, the consistently extremely low p-values across all rows (often in the order of 10e–70 or smaller) suggest that the improvements achieved by DPQEA over the baseline algorithms are statistically significant with very high confidence. This supports the claim that DPQEA performs better not only in average numerical performance but also in a statistically meaningful way.

Algorithm 2 is presented in this paper to prune the set of $m \times n$ base-learners and find the optimal subset of classifiers that constitute the best ensemble algorithm. This paper adopts DPQEA to find the best subset of base learners. To test the effect of pruning on the performance of the algorithm via the ablation scheme, the performance with and without pruning is studied. Table 7 and Table 9 presents the results for different optimisation algorithms that are used to prune the ensemble algorithm. The last row of the table shows the results of the ensemble algorithm without pruning. That is, in this case, all the $m \times n$ base-learners are used. As the data indicate, pruning the ensemble algorithm improves the performance of the ensemble. Although after pruning the number of base-learners decreases, the pruning process improves the performance. This is because the pruning process removes redundant base learners. This way, pruning improves the performance of the ensemble both in terms of computational cost and accuracy.

In Table 7, the Friedman rank of each of the algorithms is presented in the brackets. The best performance among these algorithms is reached by the proposed DPQEA, followed by QEA. Here, again, QEA shows better performance because of the nature of these algorithms, which are more suitable for binary combinatorial optimisation problems.

Table 8 and Table 10 presents a comparison between the proposed ensemble algorithm with a set of ensemble algorithms including STLR (Jurek et al., 2014), CBCA (Jurek et al., 2014), MV (Jurek et al., 2014), ABJ48 (Jurek et al., 2014), RF (Jurek et al., 2014), oRF10.1007, RoF (Zhang & Suganthan, 2015) and MPRoF-P (Zhang & Suganthan, 2015). The data in this table are averaged over 30 independent runs. In this paper, we adopt the *leave-one-country-out*, in which the data for all countries except the test countries are used to train the models, and then the models are tested on the test country. This allows a more rigorous approach where training the models is not contaminated with the test data. The numbers in the brackets show the Friedman rank of each algorithm based on the data in the table. As the data suggest, the proposed evolutionary ensemble learning (EVEL) algorithm performs best among the algorithms, followed by the MV and RF algorithms (see Tables 9 and 10). .

Figs. 4 and 5 show the growth rate of the pandemic in the UK and USA from January 2020 until May 2021. The figures show the true values of the growth rate along with their predicted values via KNN and the proposed ensemble algorithm (EVEL). We include KNN in this figure because of its good performance according to Table 2.

Table 8
The performance of different ensemble algorithms. The data are averaged over 30 runs. The numbers in the brackets show the Friedman rank of the algorithms.

	USA	Germany	UK	Italy	France
MPRoF-P (9)	1.58e−04	2.11e−04	3.28e−04	4.16e−04	3.59e−04
STLR (2)	5.97e−05	9.43e−05	1.55e−04	1.70e−04	1.60e−04
CBCA (7)	1.06e−04	1.52e−04	2.67e−04	3.64e−04	3.13e−04
MV (4)	1.04e−04	1.52e−04	2.11e−04	2.75e−04	2.53e−04
ABJ48 (6)	1.15e−04	1.47e−04	2.32e−04	2.93e−04	2.87e−04
RF (3)	7.50e−05	9.94e−05	1.68e−04	2.27e−04	2.20e−04
oRF (8)	1.37e−04	1.94e−04	2.93e−04	3.54e−04	3.47e−04
RoF (5)	9.93e−05	1.40e−04	2.58e−04	3.12e−04	2.84e−04
EVEL (1)	2.36e−05	6.23e−05	6.75e−05	9.20e−05	1.19e−04

Table 9
The performance of different optimisation algorithms for pruning the ensemble algorithm. The data are averaged over 30 runs. The numbers in the brackets show the Friedman rank of the algorithms.

	India	Brazil	S. Korea	China	Israel
DPQEA (1)	1.89e−05	2.41e−05	4.75e−05	7.18e−05	8.97e−05
QEA (3)	3.06e−05	2.70e−05	5.40e−05	8.56e−05	8.06e−05
GA (9)	3.46e−05	5.15e−05	9.79e−05	1.44e−04	1.22e−04
PSO (4)	2.75e−05	3.31e−05	7.54e−05	8.16e−05	1.10e−04
DE (2)	2.37e−05	3.74e−05	6.31e−05	7.59e−05	7.22e−05
FES (5)	2.61e−05	3.37e−05	8.63e−05	1.01e−04	8.16e−05
MASW (7)	3.27e−05	3.83e−05	6.02e−05	9.44e−05	1.15e−04
LLSO (6)	2.60e−05	4.24e−05	7.06e−05	9.12e−05	9.22e−05
RCGA (8)	2.74e−05	3.78e−05	1.07e−04	1.09e−04	1.31e−04
no pruning	7.10e−05	9.48e−05	1.77e−04	2.21e−04	2.528-04

Table 10
The performance of different ensemble algorithms. The data are averaged over 30 runs. The numbers in the brackets show the Friedman rank of the algorithms.

	India	Brazil	S. Korea	China	Israel
MPRoF-P (2)	4.87e−05	7.02e−05	9.95e−05	1.09e−04	1.37e−04
STLR (4)	6.635–05	7.43e−05	1.15e−04	1.21e−04	1.89e−04
CBCA (9)	1.30e−04	1.51e−04	2.52e−04	2.85e−04	3.69e−04
MV (8)	8.52e−05	1.08e−04	1.72e−04	1.67e−04	2.16e−04
ABJ48 (5)	6.61e−05	9.48e−05	1.32e−04	1.45e−04	1.80e−04
RF (3)	5.10e−05	6.89e−05	1.00e−04	1.10e−04	1.40e−04
oRF (7)	7.17e−05	1.07e−04	1.39e−04	1.50e−04	2.21e−04
RoF (6)	7.29e−05	8.28e−05	1.21e−04	1.47e−04	1.88e−04
EVEL (1)	1.89e−05	2.41e−05	4.75e−05	7.18e−05	8.97e−05

Other learning algorithms reach similar performance in behaviour. The proposed algorithm performs better than single-learning algorithms in two aspects. First, the predicted value of EVEL is closer to the true value of the growth rate. Second, there are fewer fluctuations in the predicted value. In the optimisation of policies, these fluctuations result in uncertainties in the fitness evaluation process. By reducing such uncertainties, the proposed ensemble algorithm provides a better estimate of the fitness of the solutions in the evolutionary optimisation of policies.

To compare the algorithms in terms of time complexity, the time it takes for each of the algorithms in seconds was measured, and the data are presented in Tables 11 and 12. Matlab 2022a was used as the implementation software, and one core of Intel(R) Core(TM) i7-6700 CPU 3.40 GHz was used as the hardware. As the data in Table 11 suggests, the base-learners take more time to train and test, compared to the base-learners, because they consist of the base-learners. The proposed algorithm takes around one minute more than the other ensemble learning algorithms because it consists of a larger number of base learners. This is because EVEL builds a pool of learning algorithms via the diversifier and pruning process.

Table 12 shows the time it takes for the diversifier algorithm to apply to any of the base learners. In our experiments, the population size is set to 10 and the number of generations is 100; thus, the diversification involves around 1000 rounds of training and testing of

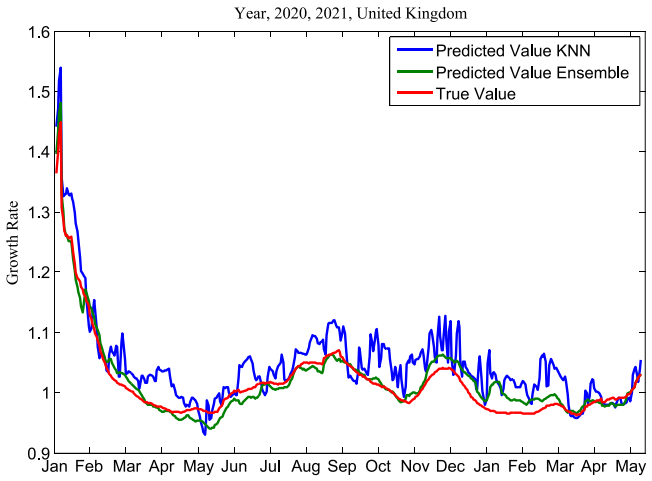

Fig. 4. The predicted and the true values of the growth rate of the number of positive COVID-19 cases in the UK. The results are for KNN and the proposed ensemble algorithm (EVEL).

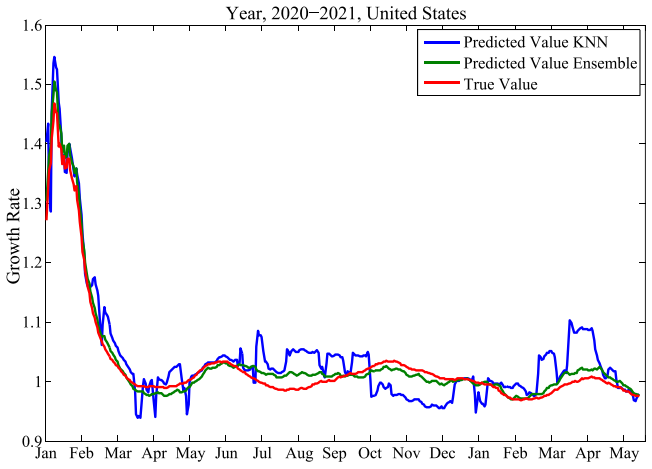

Fig. 5. The predicted and the true values of the growth rate of the number of positive COVID-19 cases in the United States. The results are for KNN and the proposed ensemble algorithm (EVEL).

the learning algorithms. This is why in this table, the time it takes for diversification is around 1000 times the training process. The time it takes for the evolutionary algorithm operators, like the crossover, mutation, etc., is negligible compared to the evaluation of the fitness function, which involves the training and testing phases. The row 'Total' in this table presents the total time it takes for the algorithm to diversify all the base learners.

The proposed algorithm also consists of the pruning process, which involves selecting a subset of base learners. In the pruning process, the evolutionary algorithms do not need to train the base learners every time the fitness is measured. The training can be performed once, and then the fitness evaluation in the pruning phase would only involve testing the base learners. It takes around 6 h for the pruning algorithm to find the best subset of the base learners.

Because the designing and training phase of the algorithms is performed only once and then, the time complexities presented in these tables are not very large. Also, note that only one core was used to perform the tasks; the process can be highly parallelised on multi-core systems, so the processing time would be reduced significantly.

Table 11
The training and testing time required for each of the algorithms. The data are in seconds.

Algorithm	Training		Testing	
	Mean	STD	Mean	STD
RBN	6.89	3.40e−02	0.55	2.63e−03
LVQ	8.46	1.90e−01	0.68	1.47e−02
PNN	0.14	1.39e−01	0.01	1.07e−02
RBE	7.16	1.35e−01	0.57	1.05e−02
CFNN	3.96	2.19e−02	0.32	1.69e−03
PRN	4.63	1.00e−02	0.37	7.76e−04
FFNN	2.60	8.79e−03	0.21	6.80e−04
GRNN	0.12	1.47e−01	0.01	1.14e−02
KNN	1.70	8.33e−01	0.14	6.44e−02
FNN	5.29	1.75e−02	0.42	1.35e−03
MPRoF-P	51.46	1.69e+00	2.95	9.91e−02
STLR	56.79	1.68e+00	1.75	6.26e−02
CBCA	56.85	1.88e+00	2.68	8.17e−02
MV	59.05	2.00e+00	2.72	1.05e−01
ABJ48	57.09	1.69e+00	2.07	6.57e−02
RF	54.32	1.61e+00	1.77	7.23e−02
oRF	42.23	1.98e+00	3.14	6.80e−02
RoF	85.22	1.79e+00	2.95	8.10e−02
EVEL	140.04	4.65e+00	3.08	6.63e−02

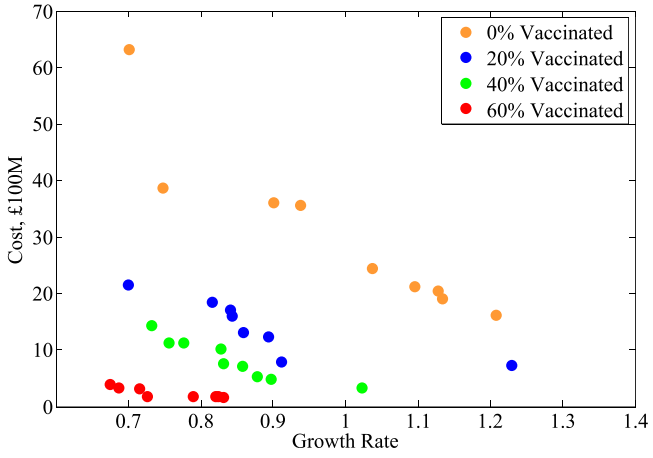

Fig. 6. The Pareto front of the policies found by the proposed algorithm, where 0, 20, 40, and 60 per cent of the population are vaccinated.

4.1. Multi-objective optimisation of government policies

In this section, we use the algorithm 3 to optimise the government policies for a given percentage of people being vaccinated. The algorithm first builds a model of the pandemic. Then, using the model to predict the growth rate of the pandemic when a particular proportion of society is vaccinated, searches through the policies and finds a set of policies that minimise the growth rate and the cost incurred on society. Because we did not have access to the data that estimates the cost of each policy to society for none of the governments, in this experiment, we use the data in Table 13 as an example to run the optimisation algorithm. Fig. 6 shows the Pareto front of the non-dominated solutions for four cases, in which 0, 20, 40, and 60 per cent of the population is vaccinated. The data indicate that when a larger number of people are vaccinated, the cost of reducing the growth rate decreases significantly. It seems also that the Pareto front for a larger percentage of vaccinations is less stretched in the horizontal axis than when fewer people are vaccinated.

Table 14 shows the performance of different optimisation algorithms in terms of Hyper-Volume. The data are averaged over 30 runs. The column denoted by "Rank" shows the Friedman rank test of each of the algorithms. As the data suggest, the best performance among the

Table 12
Time it takes by the diversifier algorithms in hour:minute:second format.

Algorithm	DPQEA	QEA	GA	PSO	DE	FES	MASW	LLSO	RCGA
RBN	01:55:40.1	01:55:42.7	01:55:13.6	01:55:29.7	01:55:28.5	01:55:18.0	01:55:35.9	01:55:46.4	01:55:58.8
LVQ	02:21:59.8	02:22:03.0	02:21:27.2	02:21:47.1	02:21:45.5	02:21:32.7	02:21:54.6	02:22:07.5	02:22:22.7
PNN	00:02:16.1	00:02:16.1	00:02:15.5	00:02:15.9	00:02:15.8	00:02:15.6	00:02:16.0	00:02:16.2	00:02:16.4
RBE	02:00:08.6	02:00:11.3	01:59:41.0	01:59:57.8	01:59:56.5	01:59:45.6	02:00:04.2	02:00:15.1	02:00:28.0
CFNN	01:06:28.3	01:06:29.8	01:06:13.0	01:06:22.3	01:06:21.6	01:06:15.5	01:06:25.8	01:06:31.9	01:06:39.0
PRN	01:17:40.4	01:17:42.1	01:17:22.5	01:17:33.4	01:17:32.5	01:17:25.5	01:17:37.5	01:17:44.6	01:17:52.9
FFNN	00:43:41.4	00:43:42.4	00:43:31.3	00:43:37.4	00:43:37.0	00:43:33.0	00:43:39.8	00:43:43.8	00:43:48.4
GRNN	00:01:56.1	00:01:56.2	00:01:55.7	00:01:55.9	00:01:55.9	00:01:55.7	00:01:56.0	00:01:56.2	00:01:56.4
KNN	00:28:29.7	00:28:30.4	00:28:23.2	00:28:27.2	00:28:26.9	00:28:24.3	00:28:28.7	00:28:31.3	00:28:34.3
FNN	01:28:48.4	01:28:50.4	01:28:28.0	01:28:40.4	01:28:39.4	01:28:31.4	01:28:45.1	01:28:53.2	01:29:02.7
Total	11:27:09.4	11:27:24.8	11:24:31.4	11:26:07.5	11:26:00.1	11:24:57.9	11:26:44.1	11:27:46.6	11:29:00.0
Pruning	05:43:38.1	05:41:40.7	05:43:24.5	05:42:41.9	05:42:07.5	05:43:06.4	05:42:38.3	05:42:03.2	05:43:54.7

Table 13
An example of the cost of each policy inflicted on society in £100M.

Policy	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}
Cost	6	2	7	5	7	5	6	7	10	8
Policy	P_{11}	P_{12}	P_{13}	P_{14}	P_{15}	P_{16}	P_{17}	P_{18}	P_{19}	P_{20}
Cost	7	2	5	4	7	4	4	6	6	8

optimisation algorithms is achieved by PSO, followed by RCGA and FES (see Table 15).

Using the second scenario described earlier, PSO is used to find a Pareto front for different percentages of cases being vaccinated. Table 17 shows the Pareto front for when 40% of the population is vaccinated. The Pareto fronts for different proportions of the vaccinated population can be found in Tables 13, 16, and 18.

5. Discussion

While the primary focus of this study is on COVID-19, the proposed modelling and optimisation framework has the potential to be applied to other infectious diseases and future pandemics. Many contagious diseases, such as seasonal influenza, measles, or novel zoonotic outbreaks, also require coordinated policy interventions and benefit from predictive modelling and resource optimisation. However, several important limitations should be acknowledged when generalising this approach. First, the epidemiological behaviour of diseases can vary significantly: factors such as incubation periods, modes of transmission, seasonal patterns, and age-specific susceptibility influence the effectiveness of different policies. As a result, the machine learning model architecture, selected algorithms, or their hyperparameters that performed best for COVID-19 may not be optimal for other diseases and would need to be re-evaluated. Second, the set of available or effective policy interventions can differ; for instance, school closures are critical for diseases affecting children disproportionately, while travel restrictions may be less relevant for endemic infections. Third, the associated social and economic costs of these policies vary by context, which may require redefining the cost functions used in the optimisation stage. Additionally, diseases with highly specific risk groups, such as illnesses primarily affecting children, the elderly, or immunocompromised individuals, require different population modelling assumptions and may involve targeted interventions. Differences in healthcare infrastructure, public compliance, and vaccination accessibility across countries further affect model transferability. Therefore, while the framework provides a generalisable structure, any application to a different disease would require careful adaptation of data inputs, epidemiological assumptions, and optimisation objectives to ensure validity and relevance.

Meta-heuristic algorithms, particularly in multi-objective settings, can be computationally intensive. However, the optimisation process in this framework is performed offline, typically only once or at infrequent intervals. This allows the use of high-performance computing resources to build the model and perform the optimisation without affecting the usability of the system in real-time decision-making. Once the optimal set of policies is identified, the model can be used rapidly and repeatedly to support policymakers, making it practical despite the initial computational overhead. Furthermore, a promising direction for future work is to adapt the system for real-time or near-real-time use, where the model is first trained on initial rounds of data collection and then incrementally retrained as new data becomes available. While retraining requires additional computational effort, it is often significantly faster than the initial model development, especially if efficient update strategies are employed. It is also important to note that for complex, high-dimensional policy optimisation problems involving conflicting objectives, such as minimising infections while reducing societal cost, meta-heuristic approaches remain among the most effective solutions available, offering flexibility and robustness that justify their computational demands.

One important limitation of the current study is the exclusion of key socio-environmental and demographic covariates–such as population density, healthcare infrastructure, age distribution, climate conditions, and urbanisation level, which are known to significantly influence the spread and severity of infectious diseases. The dataset used in this work did not include such variables, which prevented us from incorporating them into the model. Including these factors would enhance the model's causal interpretability and generalisability across regions with differing contextual conditions. However, accounting for such a broad and heterogeneous set of variables requires extensive data collection, preprocessing, and modelling decisions, which go beyond the scope of this paper. We consider such an extension to be a substantial stand-alone study, which remains for future work.

Beyond technical performance, several ethical and practical considerations must be addressed before deploying this framework in real-world policy-making. First, the dataset used in this study may contain reporting biases, missing information, or simplified representations of complex social dynamics, which could impact the validity of predictions. Second, the application of optimisation algorithms to public health decisions must be done with caution, as automated policy recommendations can raise concerns about fairness, accountability, and transparency. It is crucial to ensure that any model used to guide policy is interpretable and subject to oversight by domain experts and public health authorities. Additionally, real-world deployment would face challenges such as differences in local infrastructure, variations in population behaviour, and delays in data collection and reporting. Addressing these issues requires interdisciplinary collaboration, continuous model updating, and careful communication of uncertainty. Future work should also explore the integration of ethical frameworks and stakeholder engagement to ensure responsible and equitable use of such models in decision-making contexts.

One limitation of the current model is that it does not explicitly account for time-varying co-founders such as changes in virus variants, public behaviour, or seasonal effects, which can influence both the implementation of policies and the spread of infection. Similarly, the intensity of policy enforcement and levels of public compliance are not

Table 14
The performance of different optimisation algorithms in terms of hyper-volume. The data are averaged over 30 runs. Here, STD represents the standard deviation, and Rank shows the Friedman rank test.

Optimiser	0% vaccinated			20% vaccinated			40% vaccinated			50% vaccinated		
	Mean HV	STD	Rank	Mean HV	STD	Rank	Mean HV	STD	Rank	Mean HV	STD	Rank
PSO	30.85	0.89	1.4	11.31	0.21	1.5	5.2	0.18	1.9	4.6	0.15	1.8
GA	25.68	0.52	10.7	9.63	0.29	10.1	4.32	0.17	9.7	3.89	0.12	10.4
DE	27.14	0.41	8.5	10.08	0.27	8.2	4.51	0.31	7.5	4.09	0.18	8
ES	27.52	1.04	7.5	10.17	0.37	7.6	4.64	0.32	6.7	4.15	0.1	7.4
FES	29.44	0.4	3.3	10.84	0.56	3.9	4.92	0.28	4.2	4.41	0.14	3.3
EP	28.49	0.66	5.4	10.57	0.33	5	4.76	0.19	5.5	4.28	0.09	5.1
FEP	29.14	0.67	4.2	10.77	0.35	4.1	4.82	0.3	5.4	4.34	0.1	4.2
MASW	28.29	1.05	6.2	10.34	0.48	6.4	4.72	0.17	5.9	4.24	0.18	6.1
EDA	26.53	0.55	9.4	9.97	0.35	8.4	4.48	0.28	8.2	4.01	0.24	8.8
LLSO	27.49	0.91	7.6	10.02	0.34	8.2	4.48	0.26	8.5	4.13	0.08	7.8
RCGA	30.08	0.48	2	10.98	0.27	2.9	5.06	0.28	3	4.44	0.21	3.5

Table 15
The Pareto front where 0% of the population is vaccinated. Each column represents a solution.

G. Rate	0.7	0.75	0.9	1.1	1.13	0.9	1.0	1.1	1.2
P_1	−65	78	26	38	36	57	−78	34	−81
P_2	−29	−86	−83	11	−74	−27	56	−15	−72
P_3	−89	−52	−84	−21	44	−59	−42	−10	−67
P_4	26	−7	44	−6	−3	−4	32	32	3
P_5	1	1	3	3	0	3	3	0	1
P_6	0	0	1	1	1	0	1	0	0
P_7	20	89	10	60	32	38	69	77	21
P_8	2	0	2	2	1	1	2	2	0
P_9	2	0	1	0	2	0	1	0	2
P_{10}	1	0	0	0	1	1	1	0	2
P_{11}	4	2	3	2	3	2	0	0	2
P_{12}	0	0	0	1	0	0	2	0	0
P_{13}	2	2	0	2	2	2	0	1	0
P_{14}	3	1	3	3	4	0	1	3	0
P_{15}	−16	−56	−10	10	30	−56	19	8	37
P_{16}	0	0	2	0	0	0	1	2	2
P_{17}	2	1	2	0	1	2	3	0	2
P_{18}	0	0	2	0	1	0	1	1	0
P_{19}	−74	1	56	59	37	−19	−46	−75	−67
P_{20}	0	3	1	3	2	0	0	0	2

Table 16
The Pareto front where 20% of the population is vaccinated. Each column represents a solution.

G. Rate	0.7	0.84	0.82	0.84	1.23	0.86	0.91	0.89
P_1	−2	−41	−25	−20	−25	−37	−78	−73
P_2	−83	−67	−26	−26	−78	−65	−63	−76
P_3	−75	−71	−26	−88	−94	−1	−92	−55
P_4	3	5	−7	8	16	−3	12	−3
P_5	0	0	1	0	1	1	0	1
P_6	1	0	0	0	1	0	0	1
P_7	20	27	10	19	32	19	36	1
P_8	1	1	0	0	0	0	0	0
P_9	0	0	1	0	0	0	0	0
P_{10}	0	0	0	0	1	0	0	1
P_{11}	0	0	0	0	0	0	1	2
P_{12}	0	0	0	1	1	1	0	0
P_{13}	0	0	1	0	1	0	1	1
P_{14}	1	0	1	2	2	2	2	0
P_{15}	−35	−51	−48	−88	−63	−45	−45	−49
P_{16}	0	1	0	1	1	1	1	1
P_{17}	1	1	1	1	1	0	0	1
P_{18}	0	0	0	0	0	1	0	1
P_{19}	−47	−76	−33	−56	−94	−74	−43	−11
P_{20}	1	1	0	0	1	1	1	1

Table 17
The Pareto front where 40% of the population is vaccinated. Each column represents a solution. This is for the second scenario.

Rate	0.68	0.83	0.82	0.79	0.82	0.73	0.72	0.69
P_1	−91	−73	−48	−93	−67	−45	−82	−67
P_2	−86	−60	−54	−88	−72	−79	−85	−68
P_3	−88	−50	−100	−94	−34	−64	−53	−42
P_4	−10	3	9	1	8	12	4	−2
P_5	0	0	1	0	0	0	0	0
P_6	0	0	0	0	0	0	0	0
P_7	31	17	1	3	4	20	1	1
P_8	0	0	0	0	0	0	0	0
P_9	0	0	0	0	0	0	0	0
P_{10}	0	0	0	0	0	0	0	0
P_{11}	1	0	0	0	1	0	1	0
P_{12}	0	0	0	0	0	0	0	0
P_{13}	0	0	0	0	0	0	0	0
P_{14}	1	1	0	0	1	0	1	1
P_{15}	−84	−78	−69	−75	−94	−59	−94	−60
P_{16}	0	0	0	1	0	0	1	0
P_{17}	0	0	0	0	0	1	0	0
P_{18}	0	0	0	0	0	0	0	0
P_{19}	−55	−80	−49	−74	−74	−77	−76	−57
P_{20}	0	0	1	0	0	0	1	0

Table 18
The Pareto front where 60% of the population is vaccinated. Each column represents a solution.

G.	0.73	0.78	0.83	0.86	0.9	1.0	0.83	0.88	0.76
P_1	−80	−85	−75	−53	−80	−78	−72	−75	−51
P_2	−94	−98	−68	−73	−67	−90	−84	−68	−73
P_3	−87	−91	−85	−64	−54	−56	−78	−53	−75
P_4	−6	2	−8	0	4	3	2	2	−5
P_5	0	0	0	0	0	0	0	0	0
P_6	0	0	0	0	0	0	0	0	0
P_7	8	14	19	16	10	16	0	3	1
P_8	0	0	0	0	0	0	0	0	0
P_9	0	0	0	0	0	0	0	0	0
P_{10}	0	0	0	0	0	0	0	0	0
P_{11}	1	0	0	1	0	0	0	0	1
P_{12}	0	0	0	0	0	0	0	0	0
P_{13}	0	0	0	0	0	0	0	0	0
P_{14}	0	0	1	0	0	1	0	0	1
P_{15}	−73	−69	−64	−68	−67	−70	−88	−97	−67
P_{16}	0	0	0	0	0	0	0	0	0
P_{17}	0	0	0	0	0	0	0	0	0
P_{18}	0	0	0	0	0	0	0	0	0
P_{19}	−56	−87	−71	−68	−93	−59	−81	−52	−56
P_{20}	0	0	0	0	0	0	0	0	0

captured in the available dataset, even though they can vary significantly over time and across regions. As a result, the model assumes a static interpretation of policy inputs and may not fully reflect the complexity of real-world dynamics. Incorporating such temporal and behavioural variations would likely improve the causal reliability and predictive accuracy of the model. Future work will address this by

integrating additional data sources such as mobility trends, compliance indicators, and genomic surveillance data, and by extending the framework to explicitly model temporal dependencies and policy adaptation over time.

The current study includes a performance analysis of training and optimisation times based on the available dataset, as shown in Tables 11 and 12. The evolutionary optimisation process for constructing the ensemble model takes approximately 15 h, while training the final model requires about two minutes, and testing takes less than a second. These times are acceptable for offline optimisation and deployment scenarios. However, the scalability of the proposed method to significantly larger datasets or real-time data streams remains an open question. Since training times in machine learning generally scale linearly with data size, we expect similar trends here, but validating this requires further empirical testing with larger datasets. Future work will focus on benchmarking the method against growing data volumes and exploring strategies for incremental or online learning, where the system can adapt as new data becomes available without full retraining. Additionally, techniques for parallelisation or approximation during the evolutionary search phase could be investigated to further reduce computational cost and improve feasibility for real-time or resource-constrained environments.

One important consideration in applying the proposed framework is the robustness of the model to missing or incomplete data. In practice, the dataset used in this study already suffers from various forms of incompleteness and reporting bias. For example, the reported number of infected cases is an underestimation due to asymptomatic infections, limited testing availability, and differences in testing protocols across countries. Many individuals with symptoms may not get tested, and in lower-income regions, the high cost of testing leads to under-reporting. Furthermore, policy records themselves are not always recorded with complete detail or consistency. Despite these limitations, the proposed model has been designed to function effectively with such imperfect data, and the results presented are based on these real-world conditions. While studying the algorithm under fully complete and accurate data would provide further insight, access to such datasets is extremely limited or unrealistic at a global scale. Future work may explore the integration of imputation techniques or uncertainty modelling to further enhance the model's robustness to missing data.

6. Conclusion

This paper proposes an evolutionary optimisation of government policies when a particular percentage of society is vaccinated. The proposed method uses an ensemble algorithm to build a model of the pandemic, which gets as input the set of policies and percentage of people vaccinated, and predicts as output the growth rate of the pandemic. Data from a well-known, daily updated source is collected and used to build the model. The model is then used in an evolutionary algorithm that searches through all possible policies and finds the one that causes the lowest growth rate of the pandemic. The proposed ensemble learning algorithm is designed via an evolutionary diversifier which gets a learning algorithm and, by choosing different subsets of features, generates different base-learners that are diverse. Using different learning algorithms, several base learners are created. Then, an evolutionary algorithm chooses an optimal subset of these base learners to build the final ensemble algorithm. Experimental analysis suggests that the proposed ensemble algorithm improves the performance of the base learners. This model is then used as a fitness function in a multi-objective optimisation algorithm to find the optimal set of policies.

The data from the 1st of January 2020, until mid-May 2021 (the date of the writing of this paper) were used to build the model and test the algorithms. There are some shortcomings in the data that we hope will be alleviated as time progresses. At the time of the writing of this paper, not all countries have started mass vaccination, so not much data is available regarding the vaccination. Also, not all countries release accurate data about the policies they have taken and the number of positive cases. The data about the number of cases is very noisy, which stems from different complications. One complication is that the number of cases reported by the government is rarely correct. This is due to the difficulties associated with testing (including costs and planning) and also the fact that many cases are asymptomatic and remain under the radar. While these shortcomings remain, the models of the pandemic are not very accurate.

Although our best efforts were spent on building the algorithms in this paper, a great deal of work and difficulties remain for future work to be resolved. First, new strains of the virus appear frequently, and some of them have changed enough to show different behaviour in terms of infection rate and their response to vaccination. How the appearance of new strains affects the performance of the existing models and how to develop models that take this into account is a topic of research for future work. Although this requires solid data collection about the number of cases of each strain around the world, which is still not available in the data sources. Second, it is not only the number of vaccinated people that affects the behaviour of the pandemic. Also, the number of people who have been infected recently is a matter of importance, as they bear a form of immunity. Because not much data was available about the number of people who are immune due to infection, we did not consider the effect in this paper. Data about the antibody tests can provide more accurate indications about the level of immunity in populations; however, such data is still not available in most countries. Third, different types of vaccines have been administered in different countries or even within the same country. At the moment, the data released by the governments does not include detailed information about the type of vaccines administered. Because of the differences between the efficacy of different types of vaccines against the virus and its different strains, a model that only considers the percentage of vaccinated people would not perform very accurately. In the future, if such data is available, types of vaccines and their effect on different variants of the virus should be taken into account.

The proposed algorithm, like many other ensemble learning algorithms, is computationally complex as it involves the training, diversification, and pruning of the base learners. Such a task is more time-consuming than training a single learning algorithm. However, we believe that the improvement in terms of performance and robustness that the proposed algorithm offers outweighs its cost.

In this paper, we considered the government policies and the vaccination level as the features to build the model. However, many other factors, including weather, political systems, religion, geography, income, nutrient intake, race, etc., affect the behaviour of the pandemic. Collecting data about these factors and performing statistical analysis and feature selection to find out which factors should and can be incorporated into the models is another line of research that remains for future work.

One might argue that now the vaccines have been developed, and thus research like this will get outdated when all people are fully vaccinated. The answer is that this pandemic may be over soon, but the grounds on which this research was conducted will remain. First, although the developed countries have reached a good level of vaccination, it takes a while until all the countries, including the developing countries, get fully vaccinated or reach a considerable level of immunity, and until then, there is a need for research like this to tackle the problem. Third, the appearance of new strains of the virus may make the vaccines partially effective; thus, even if a population is fully vaccinated or immune due to infection, the pandemic will behave as if only a certain percentage of the population is vaccinated. This probably continues for a long time, or even forever. Finally, this pandemic will not be the last one that humanity faces. New types of viruses or bacteria can cause new pandemics in the future. It is expected to see more pandemics arise with higher frequency because of the way we have defined our relationship with the environment and

wildlife (Young et al., 2014). Therefore, research like this will help to better understand and tackle future pandemics.

We believe implementing a successful set of policies against any pandemic would involve employing computational intelligence algorithms. The method presented in this paper is particularly designed for this matter, to help policy-makers in their decision-making tasks. Such work not only can be useful during the pandemic but also can be used for estimating the costs a future pandemic may cause.

The approach presented in this work was implemented, tested, and studied on a specific application. However, this remains an open question of whether and to what extent such an approach improves the performance of the modelling algorithms for other applications. While this remains for future work to investigate, the author believes that the proposed algorithm should improve the performance for many applications, because it is shown in the literature that ensemble algorithms outperform single algorithms if there is diversity among the base learners. Because the approach in this paper is to improve diversity, the approach should work for many other applications.

Declaration of competing interest

All authors have participated in (a) conception and design, or analysis and interpretation of the data; (b) drafting the article or revising it critically for important intellectual content; and (c) approval of the final version.

This manuscript has not been submitted to, nor is under review at, another journal or other publishing venue.

The authors have no affiliation with any organisation with a direct or indirect financial interest in the subject matter discussed in the manuscript

Data availability

Data will be made available on request.

References

Alzubi, O. A., Alzubi, J. A., Alweshah, M., Qiqieh, I., Al-Shami, S., & Ramachandran, M. (2020). An optimal pruning algorithm of classifier ensembles: dynamic programming approach. *Neural Computing and Applications*, *32*(20), 16091–16107.

Chen, H., & Yao, X. (2006). Evolutionary multiobjective ensemble learning based on Bayesian feature selection. In *2006 IEEE international conference on evolutionary computation* (pp. 267–274). http://dx.doi.org/10.1109/CEC.2006.1688318.

Cheng, B., Wu, W., Tao, D., Mei, S., Mao, T., & Cheng, J. (2020). Random cropping ensemble neural network for image classification in a robotic arm grasping system. *IEEE Transactions on Instrumentation and Measurement*, *69*(9), 6795–6806. http://dx.doi.org/10.1109/TIM.2020.2976420.

Conlon, A., Ashur, C., Washer, L., Eagle, K. A., & Bowman, M. A. H. (2021). Impact of the influenza vaccine on COVID-19 infection rates and severity. *American Journal of Infection Control*.

Dai, Q., Ye, R., & Liu, Z. (2017). Considering diversity and accuracy simultaneously for ensemble pruning. *Applied Soft Computing*, *58*, 75–91.

Deng, L., Wang, S., Qiao, L., & Zhang, B. (2018). DE-RCO: Rotating crossover operator with multiangle searching strategy for adaptive differential evolution. *IEEE Access*, *6*, 2970–2983.

Dietterich, T. G. (2000). Ensemble methods in machine learning. In *Multiple classifier systems* (pp. 1–15). Berlin, Heidelberg: Springer Berlin Heidelberg.

Hayashi, Y., & Iiduka, H. (2018). Optimality and convergence for convex ensemble learning with sparsity and diversity based on fixed point optimization. *Neurocomputing*, *273*, 367–372.

Jin, Y., & Branke, J. (2005). Evolutionary optimization in uncertain environments-a survey. *IEEE Transactions on Evolutionary Computation*, *9*(3), 303–317.

Jurek, A., Bi, Y., Wu, S., & Nugent, C. D. (2014). Clustering-based ensembles as an alternative to stacking. *IEEE Transactions on Knowledge and Data Engineering*, *26*(9), 2120–2137.

Kolárik, M., Sarnovský, M., & Paralič, J. (2021). Diversity in ensemble model for classification of data streams with concept drift. In *2021 IEEE 19th world symposium on applied machine intelligence and informatics* (pp. 000355–000360).

Li, D., & Wen, G. (2018). MRMR-based ensemble pruning for facial expression recognition. *Multimedia Tools and Applications*, *77*(12), 15251–15272.

Li, D., Wen, G., Li, X., & Cai, X. (2019). Graph-based dynamic ensemble pruning for facial expression recognition. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, *49*(9), 3188–3206.

Lim, P., Goh, C. K., & Tan, K. C. (2017). Evolutionary cluster-based synthetic oversampling ensemble (ECO-ensemble) for imbalance learning. *IEEE Transactions on Cybernetics*, *47*(9), 2850–2861. http://dx.doi.org/10.1109/TCYB.2016.2579658.

Liu, H., & Chen, S. (2019). Multi-perspective creation of diversity for image classification in ensemble learning context. In *2019 international conference on machine learning and cybernetics* (pp. 1–6). http://dx.doi.org/10.1109/ICMLC48188.2019.8949189.

Mao, S., Chen, J. W., Jiao, L., Gou, S., & Wang, R. (2019). Maximizing diversity by transformed ensemble learning. *Applied Soft Computing*, *82*, Article 105580.

Max Roser, E. O. O., & Hasell, J. (2020). Coronavirus pandemic (COVID-19). *Our World in Data*, https://ourworldindata.org/coronavirus.

Mininno, E., Cupertino, F., & Naso, D. (2008). Real-valued compact genetic algorithms for embedded microcontroller optimization. *IEEE Transactions on Evolutionary Computation*, *12*(2), 203–219.

Molina, D., Lozano, M., Sánchez, A., & Herrera, F. (2011). Memetic algorithms based on local search chains for large scale continuous optimisation problems: MA-SSW-chains. *Soft Computing*, *15*, 2201–2220.

Najaran, M. H. T. (2023a). An evolutionary ensemble learning for diagnosing COVID-19 via cough signals. *Intelligent Medicine*.

Najaran, M. H. T. (2023b). A genetic programming-based convolutional deep learning algorithm for identifying COVID-19 cases via X-ray images. *Artificial Intelligence in Medicine*, *142*, Article 102571.

Najaran, M. H. T., & Tootounchi, M. R. A. (2020). Probabilistic optimization algorithms for real-coded problems and its application in latin hypercube problem. *Expert Systems with Applications*, *160*, Article 113589.

Oliveira, D. V., Cavalcanti, G. D., & Sabourin, R. (2017). Online pruning of base classifiers for dynamic ensemble selection. *Pattern Recognition*, *72*, 44–58.

Polikar, R. (2006). Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine*, *6*(3), 21–45.

Pratama, M., Pedrycz, W., & Lughofer, E. (2018). Evolving ensemble fuzzy classifier. *IEEE Transactions on Fuzzy Systems*, *26*(5), 2552–2567. http://dx.doi.org/10.1109/TFUZZ.2018.2796099.

Rakshit, P., Konar, A., & Das, S. (2017). Noisy evolutionary optimization algorithms–a comprehensive survey. *Swarm and Evolutionary Computation*, *33*, 18–45.

Sun, T., & Zhou, Z. H. (2018). Structural diversity for decision tree ensemble learning. *Frontiers of Computer Science*, *12*(3), 560–570.

Tayarani, M. (2020). Applications of artificial intelligence in battling against covid-19: A literature review. *Chaos, Solitons & Fractals*.

Tayarani-N, M. H., & Akbarzadeh-T, M. (2014). Improvement of the performance of the quantum-inspired evolutionary algorithms: structures, population, operators. *Evolutionary Intelligence*, *7*(4), 219–239.

Tayarani-N., M., Yao, X., & Xu, H. (2015). Meta-heuristic algorithms in car engine design: A literature survey. *IEEE Transactions on Evolutionary Computation*, *19*(5), 609–629. http://dx.doi.org/10.1109/TEVC.2014.2355174.

Wang, J., Wang, Q., Zhang, H., Chen, J., Wang, S., & Shen, D. (2019). Sparse multiview task-centralized ensemble learning for ASD diagnosis based on age- and sex-related functional connectivity patterns. *IEEE Transactions on Cybernetics*, *49*(8), 3141–3154. http://dx.doi.org/10.1109/TCYB.2018.2839693.

Xia, X., Lin, T., & Chen, Z. (2018). Maximum relevancy maximum complementary based ordered aggregation for ensemble pruning. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, *48*(9), 2568–2579.

Yang, L. (2011). Classifiers selection for ensemble learning based on accuracy and diversity. *Procedia Engineering*, *15*, 4266–4270, CEIS 2011.

Yang, Q., Chen, W., Deng, J. D., Li, Y., Gu, T., & Zhang, J. (2018). A level-based learning swarm optimizer for large-scale optimization. *IEEE Transactions on Evolutionary Computation*, *22*(4), 578–594. http://dx.doi.org/10.1109/TEVC.2017.2743016.

Yang, X., Xu, Y., Quan, Y., & Ji, H. (2020). Image denoising via sequential ensemble learning. *IEEE Transactions on Image Processing*, *29*, 5038–5049. http://dx.doi.org/10.1109/TIP.2020.2978645.

Yang, P., Yoo, P. D., Fernando, J., Zhou, B. B., Zhang, Z., & Zomaya, A. Y. (2014). Sample subset optimization techniques for imbalanced and ensemble learning problems in bioinformatics applications. *IEEE Transactions on Cybernetics*, *44*(3), 445–455. http://dx.doi.org/10.1109/TCYB.2013.2257480.

Yao, X., & Liu, Y. (1997). Fast evolution strategies. *Control and Cybernetics*, *26*, 467–496.

Young, H. S., Dirzo, R., Helgen, K. M., McCauley, D. J., Billeter, S. A., Kosoy, M. Y., Osikowicz, L. M., Salkeld, D. J., Young, T. P., & Dittmar, K. (2014). Declines in large wildlife increase landscape-level prevalence of rodent-borne disease in africa. *Proceedings of the National Academy of Sciences*, *111*(19), 7036–7041. http://dx.doi.org/10.1073/pnas.1404958111.

Zhang, Y., Cao, G., & Li, X. (2021). Multiview-based random rotation ensemble pruning for hyperspectral image classification. *IEEE Transactions on Instrumentation and Measurement*, *70*, 1–14. http://dx.doi.org/10.1109/TIM.2020.3011777.

Zhang, S., Chen, Y., Zhang, W., & Feng, R. (2021). A novel ensemble deep learning model with dynamic error correction and multi-objective ensemble pruning for time series forecasting. *Information Sciences*, *544*, 427–445.

Zhang, L., & Suganthan, P. N. (2015). Oblique decision tree ensemble via multisurface proximal support vector machine. *IEEE Transactions on Cybernetics*, *45*(10), 2165–2176.

Zhu, Z., Wang, Z., Li, D., Zhu, Y., & Du, W. (2020). Geometric structural ensemble learning for imbalanced problems. *IEEE Transactions on Cybernetics*, *50*(4), 1617–1629. http://dx.doi.org/10.1109/TCYB.2018.2877663.

Mohammad- H. Tayarani- N. received his Ph.D. degree from the University of Southampton, Southampton, U.K, in 2013. Then he worked as a research fellow at the University of Birmingham, Birmingham, UK, and the University of Glasgow, Glasgow, UK. He currently holds a fellowship at the University of Hertfordshire, Hatfield, UK. His main research interests include evolutionary algorithms, machine learning, and fractal image compression.