A Novel Ensemble Machine Learning and an Evolutionary Algorithm in Modeling the COVID-19 Epidemic and Optimizing Government Policies

Mohammad-Hassan Tayarani-Najaran

***Abstract*—The spread of the COVID-19 disease has prompted a need for immediate reaction by governments to curb the pandemic. Many countries have adopted different policies and studies are performed to understand the effect of each of the policies on the growth rate of the infected cases. In this article, the data about the policies taken by all countries at each date, and the effect of the policies on the growth rate of the pandemic are used to build a model of the pandemic's behavior. The model takes as input a set of policies and predicts the growth rate of the pandemic. Then, a population-based multiobjective optimization algorithm is developed, which uses the model to search through the policy space and finds a set of policies that minimize the cost induced to the society due to the policies and the growth rate of the pandemic. Because of the complexity of the modeling problem and the uncertainty in measuring the growth rate of the pandemic via the models, an ensemble learning algorithm is proposed in this article to improve the performance of individual learning algorithms. The ensemble consists of ten learning algorithms and a metamodel algorithm that is built to predict the accuracy of each learning algorithm for a given data record. The metamodel is a set of support vector machine (SVM) algorithms that is used in the aggregation phase of the ensemble algorithm. Because there is uncertainty in measuring the growth rate via the models, a landscape smoothing operator is proposed in the optimization process, which aims at reducing uncertainty. The algorithm is tested on open access data online and experiments on the ensemble learning and the policy optimization algorithms are performed.**

***Index Terms*—Ensemble learning, epidemiology COVID-19, evolutionary algorithms, optimization, policy making.**

I. Introduction

IT IS a great challenge for governments around the world to find the optimal policies to curb the spread of the pandemic. Stricter measures can control the pandemic, thus reducing the number of cases and deaths, at the cost of high economic impact. In this sense, it is crucial to develop methods that can find the optimal policies that reduce the number of cases with a minimal economic impact on society. Developing such methods requires the design of algorithms that can predict the evolution of the pandemic as it allows governments to develop strategic planning to curb the effects. In order to find the optimal government policies against the spread of the virus, in this article, we propose an evolutionary algorithm that searches through different policies and finds the best policy that minimizes the spread of the virus and the cost imposed on society.

In order to implement the optimization algorithm, there should be a model of the epidemic that can take as input the policies (optimization parameters) and produce as output the spreading rate of the virus. This model is to be used as the fitness function for the optimization algorithm. To build such a model, an ensemble learning algorithm is proposed in this article.

One challenge in using optimization algorithms via surrogate models is the uncertainty in the fitness evaluation. Known as approximation uncertainty, this type of uncertainty occurs when a model is used to approximate or estimate the fitness function [1]. Such uncertainty affects the optimization process and should be managed. Because there is uncertainty in the evaluated value of growth rate due to a set of policies, we propose landscape smoothing operators for fitness evaluation of the optimization algorithm. The proposed method removes high fluctuations in the fitness evaluation process that are usually due to uncertainty.

A. Previous Work

Ensemble learning improves the performance by combining algorithms that complement one another [2]–[5]. The ensemble improves performance by providing diversity and accuracy [6]. In order to reach diversity, two main approaches called heterogeneous and homogeneous are usually employed. Diversity is achieved if a set of classifiers is used that misclassify different instances [7]. In homogeneous approaches, the randomness is injected into the training phase of the classification, which is achieved by methods like manipulating the feature set or the learning algorithms. Among these approaches are Bagging [8], in which the distribution of the training data is changed to achieve different training sets, random subspace [9], which randomly selects subsets of features to build diversity, and the diversification of algorithm parameters scheme in which the diversity is incorporated into the learning algorithms [10].

In ensemble learning, two main approaches are the Bagging [8] and AdaBoost [11] algorithms. In order to achieve

Manuscript received 30 June 2021; revised 10 November 2021; accepted 9 January 2022. Date of publication 4 February 2022; date of current version 16 September 2022. This article was recommended by Associate Editor J. A. Lozano.

The author is with the School of Computer Science, University of Hertfordshire, Hatfield AL10 9AB, U.K. (e-mail: tayaranm@herts.ac.uk).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TSMC.2022.3143955.

Digital Object Identifier 10.1109/TSMC.2022.3143955

diversity, these two approaches strategically generate classifiers by manipulating the training data. Bagging approaches perform this by using different datasets. Theoretical study of bagging approaches can be found in [12] and [13]. In [14], a variation of Bagging for large datasets is proposed, which is called *Pasting small votes*. Two versions of these algorithms were proposed, one which is called *Rvotes* creates the data subsets at random and the other, called *Ivotes* creates consecutive datasets based on important data instances that promote diversity. An approach called attribute bagging is proposed in [15], which is a wrapper algorithm and establishes an appropriate attribute subset size. The bootstrap sampling is integrated in [16] with more advanced methods of feature selection, which are based on an evaluation of the relationship between the features and the target class. In order to improve the performance of the Bagging algorithm, a method is introduced in [17] that considers the diversity of classification margins in feature subspaces. To do so, the task is converted into an optimization problem of finding the best weights of feature subspaces.

Dynamic classifier selection and classifier fusion are two main approaches that are used in the literature. Dynamic classifier selection approaches try to predict the accuracy of the classifiers to choose the output of what classifier should be used as output. In classifier fusion methods, the output of classifiers is aggregated to reach a consensus.

Ensemble approaches have been used in a number research to tackle the COVID-19 problems. In [18], an ensemble deep learning algorithm is proposed for the detection of COVID-19 via CT images. An automatic detection of COVID-19 via X-ray images is presented in [19], which utilizes an ensemble of deep CNN. An ensemble of learning algorithms is used in [20] to diagnose COVID-19 cases via routine blood test. To study vaccine efficacy trials, in [21], an ensemble learning algorithm is adopted. An optimization algorithm is proposed in [22] to build a sparse ensemble algorithm to predict the evolution of COVID-19.

Machine learning algorithms have been used in some research to build models of the pandemic. An ensemble empirical mode decomposition is proposed in [23], which is combined with ANN to predict the pandemic. In another work [24], in order to predict the pandemic in Egypt, statistical and AI-based approaches are combined. In this work, ARIMA and nonlinear autoregressive artificial neural networks (NARANNs) are integrated. In [25], an ensemble of neural networks is presented to build a model of the pandemic in Mexico. Then, a fuzzy logic system is employed to reach consensus among the response of these neural predictors. Neural networks with long short-term memory networks are combined in [26] to build a model to forecast the pandemic. In [27], some neural network forecasting methods, including multilayer perceptron, neural network autoregressive, and extreme learning machine, are used to study the effectiveness of public health measures on the epidemic. The model is used to predict the number of active, confirmed, recovered, death, and daily new cases in Jakarta and Java. A machine learning-based time-series prediction model using the FbProphet model is used in [28] to predict the epidemic curve in Brazil, Russia, India, Peru, and Indonesia. In [29], several regression analysis models are used to analyze the epidemiological data of Egypt and predict the pandemic trend. A cloud computing platform is used in [30] to develop a machine learning algorithm that predicts the threat of COVID-19 cases in countries worldwide. The system provides a real-time prediction of the growth behavior of the epidemic. In [31], a metaanalysis of the current state of the art of the AI approaches to tackle COVID-19 is presented. A random forest model is used in [32] to estimate the number of cases in 190 countries. A comprehensive review of the recent research on the applications of AI in battling against COVID-19 can be found in [33].

While these research show promising results in modeling the pandemic, they do not take the set of government policies as the input for the models. In this sense, these modeling algorithms do not approach the problem of using the models to find an optimal set of policies to reduce the number of positive cases. The current paper targets the modeling of the behavior of the pandemic with the aim of providing a practical approach for policy makers.

The remainder of this article is organized as follows. The proposed ensemble learning algorithm is discussed in Section II. Section III describes the optimization algorithm proposed in this article. The experimental studies are performed in Section IV and finally, Section V concludes this article.

II. Proposed Ensemble Learning Algorithm

In this article, we build a model of the pandemic to predict the growth rate for a given set of policies. Then, an evolutionary algorithm is used to search through the set of all policies to minimize the growth rate and the cost induced to society. For many reasons, the task of modeling the behavior of the pandemic is very hard for the existing modeling algorithms. First, the behavior of the pandemic is very complicated, which is affected by a huge number of unpredictable factors. This article tries to model the behavior of pandemic based on government policies. Surely, there are many other factors that affect the behavior of the pandemic and are extremely hard to take into account when building the models. Some examples include cultural factors, existing immunities within the society, the density of population, age demography, etc. Second, the data are not complete, many countries do not collect data, and it is almost impossible to know the true number of cases in a country. This is because testing is costly and a significant number of cases remain asymptomatic thus undetected. In this respect, there is a need to improve the performance of the algorithms to better predict the growth rate.

One inevitable consequence of using surrogate models to evaluate the solutions is the uncertainty they suffer from in the estimation of the growth rate. Reducing this uncertainty is crucial as it can mislead the search process. There is much research on how to manage such uncertainty [34]. Most of these works use different averaging techniques to reduce the noise by resampling the fitness function several times. In our problem, however, recalculating the growth rate via the model

TABLE I
PAIRWISE COMPARISON BETWEEN THE LEARNING ALGORITHMS. THIS SHOWS THE PERCENTAGE OF TIMES AN ALGORITHM PERFORMS BETTER THAN THE OTHER. THIS IS WHERE THE DATA FOR ALL COUNTRIES ARE USED FOR TRAINING AND THE DATA FOR U.K. ARE USED FOR TESTING. THE DATA ARE AVERAGED OVER 50 RUNS

	RBN	LVQ	PNN	RBE	CFNN	PRN	FFNN	GRNN	KNN	FNN
RBN	-	47.96	51.49	52.08	47.76	10.86	38.17	58.58	60.07	67.91
LVQ	52.04	-	28.05	66.56	58.84	20.29	30.21	58.47	60.35	64.28
PNN	48.51	71.95	-	66.79	59.33	13.99	34.33	61.77	61.94	66.84
RBE	47.92	33.44	33.21	-	58.21	27.16	32.46	61.57	62.31	68.66
CFNN	52.24	41.16	40.67	41.79	-	21.56	27.99	61.19	60.45	69.03
PRN	89.14	79.71	86.01	72.84	78.44	-	63.02	66.43	65.14	66.89
FFNN	61.83	69.79	65.67	67.54	72.01	36.98	-	69.4	70.15	75.37
GRNN	41.42	41.53	38.23	38.43	38.81	33.57	30.6	-	64.18	63.02
KNN	39.93	39.65	38.06	37.69	39.55	34.86	29.85	35.45	-	64.93
FNN	32.09	35.72	33.16	31.34	30.97	33.11	24.63	36.98	35.07	-

would not reduce the uncertainty as the modeling algorithms produce deterministic values.

To overcome these challenges, we propose an ensemble algorithm in this article. Ensemble algorithms usually improve the performance of individual learning algorithms by benefiting from the advantages of base learners. Using many learning algorithms instead of one not only improves the performance in terms of accuracy but also injects diversity into the output of the models, which is required in the noise reduction process.

Ensemble learning is when multiple learning algorithms are combined to solve a problem. In this article, we propose an ensemble algorithm to model the data. The base learning algorithms used in the proposed ensemble algorithm are feed-forward neural network (FNN), radial basis network (RBN), learning vector quantization (LVQ) neural network, probabilistic neural networks (PNN), exact radial basis network (RBE), cascade-forward neural network (CFNN), pattern recognition network (PRN), function fitting neural network (FFNN), generalized regression neural network (GRNN), and K-nearest neighbor (KNN). For all these algorithms, the MATLAB 2016a implementations are used. Unless specifically mentioned, the default parameters of the algorithms are used. For FNN, the number of hidden layers is set to 1, with 20 hidden nodes, for RBN, GRNN, and PNN, the spread parameter is set to 0.5, for LVQ, the number of nodes in the hidden layer is set to 10 and the number of epoch to 50, for CFNN, the number of nodes in the hidden layer is set to 10, for FFNN, the number of hidden nodes is set to 30, and for KNN, the number of neighbors k is set to 5.

In the proposed algorithm, the accuracy of the base learners is estimated via a metalearning algorithm and final results among the base learning algorithms are achieved based on the estimated accuracy of these algorithms. The proposed metalearning algorithm consists of a number of support vector machine (SVM) algorithms that are trained to model the accuracy of each base learner. A pairwise comparison between the accuracy of each algorithm for each data record is performed and for each pair of base learners, an SVM is trained to predict for which data records each of the learners will outperform the other. These sets of SVMs provide by combining a number of learning algorithms, the proposed ensemble algorithm outperforms the individual base learners. The accuracy of each base learner is modeled based on the data in the training set.

Fig. 2 in the supplementary material presents how is some regions in the feature space one algorithm outperforms another one. This suggests that there are some regions in feature space in which one algorithm outperforms other algorithms. If for a particular data point the performance of the learning algorithms is predicted, then in the aggregation step, the output of the algorithm with better performance is better to be used as the final output of the ensemble algorithm.

In building an ensemble of classifiers, diversity should be achieved, and it is achieved when a set of classifiers is used that misclassify different instances [7]. In an ensemble of modeling algorithms, diversity is achieved if there are instances for which each algorithm outperforms the others. As presented in Fig. 2 in the supplementary material, when comparing two learning algorithms on test data, there are instances that GRNN outperforms KNN and there are instances that the opposite occurs. In order to show the diversity in the base learners in this article, Table I performs a pairwise comparison between the algorithms to show in what percentage of instances each algorithm outperforms the other. In this table, the number at a cell shows the percentage of the times in which the learning algorithm labeling the column outperforms the learning algorithm that labels the row. That is, for example, in 47.96% of times, LVQ outperforms RBN and in 52.04% of times, the opposite happens. In order to generate these data, the modeling algorithms are used to predict the growing rate of the pandemic for all the test data records. Then, to compare two learning algorithms, the outputs of the two learning algorithms for each data record are compared and the percentage of the data records for which a learning algorithm performs better than the other is reported in the table. The data for all countries except the U.K. are used to train the learning algorithms and the data for the U.K. are used to test. The data are averaged over 50 independent runs, where for each run the learning algorithms are trained and tested independently.

In some cases, some learning algorithms perform much better than some others, for example, in 71.95% of the times, LVQ outperforms PNN. Note that although in majority of times one algorithm outperforms the other one, a smart ensemble of these algorithms can improve the performance further. It can be achieved by knowing in what cases which algorithm provides better prediction.

In order to study if SVM is capable of identifying where in the feature space each learning algorithm outperforms

TABLE II
PERFORMANCE OF SVM IN PREDICTING THE ACCURACY OF THE LEARNING ALGORITHMS. THE DATA ARE AVERAGED OVER 50 RUNS

	RBN	LVQ	PNN	RBE	CFNN	PRN	FFNN	GRNN	KNN
LVQ	89.25	-	-	-	-	-	-	-	-
PNN	89.29	97.33	-	-	-	-	-	-	-
RBE	86.74	87.34	87.61	-	-	-	-	-	-
CFNN	74.53	78.71	79.06	78.78	-	-	-	-	-
PRN	90.48	91.6	91.12	91.44	92.72	-	-	-	-
FFNN	76.95	79.18	79.1	80.98	64.52	90.94	-	-	-
GRNN	80.94	83.14	83.12	83.75	64.42	94.68	72.55	-	-
KNN	81.21	82.78	82.96	83.68	63.29	94.42	71.21	67.01	-
FNN	79.02	78.08	77.99	78.69	59.64	93.23	66.13	66.31	65.09

the other one, Table II shows the performance of SVM in predicting the performance of the learning algorithms. The data show the accuracy of SVM in predicting which of the algorithms performs better for test data in a pairwise comparison. To generate these data, the data for all countries are divided into 1/2 training, X and 1/2 testing data, T. The training data X are used to train the ten classifiers used in this article. The testing data T are then split into 2/3, W for training the SVMs and 1/3, V testing the SVMs. For each pair of classifiers, an SVM is trained to predict for which data records which of the algorithms outperforms the other. To do so, the learning algorithms are tested on W and a pairwise comparison is performed on the performance of all the learning algorithm on all the data records in W. Then, an SVM is trained on the pairwise comparison on all the data in W to learn which of the learning algorithm outperforms the other one for each of the data records. This way, the SVM is trained to learn where in the feature space which of the algorithms performs better. The SVMs are then tested on the data V and the results are reported in the table. The data are averaged over 50 independent runs, where the sets X, T, W, and V are chosen independently randomly for each run.

As presented in Table II with very good accuracy, SVM is capable of predicting which algorithm performs better. When using an ensemble of learning algorithms, having a prediction of which of the algorithms is more likely to provide more accurate results can help to decide the output of which of the learning algorithms is better to be used as output.

The idea in the proposed ensemble method is that different learning algorithms perform differently in different regions in the feature space. Therefore, for a given data point, the performance of the algorithms at the data point is estimated, and the output of the algorithm with higher estimated performance can be chosen as the output. In the proposed algorithm, we use a number of learning algorithms to model the growth rate of the epidemic. Then, we use the SVM algorithm to model the performance of the algorithms in different regions.

The proposed ensemble learning method is presented in Algorithm 1. A description of the proposed algorithm is as follows. First, the data are divided into training and testing sets. In steps 2–12, the training set is used to build a model that predicts the performance of different learning algorithms in different regions in the feature space. In step two, the training data are partitioned into four subsets. Three of the partitions are used as training and one partition as the test set. This

Algorithm 1: Training Phase of the Proposed Ensemble Algorithm

begin
1. Divide the data into train and test sets
2. Partition the train set, X into four partitions of equal size, X^i, $i = 1 \dots 4$
3. For $i = 1 \rightarrow 4$
 begin
4. For $l = 1 \rightarrow |L|$ train the l-th modelling algorithm, L_l, using $\bigcup_{\forall j \neq i} X^j$
5. For $l = 1 \rightarrow |L|$
6. For $k = 1 \rightarrow |X^i|$ find the output of L_l for X^i_k, $L_l(X^i_k)$
7. For $l = 1 \rightarrow |L|$
8. For $m = l + 1 \rightarrow |L|$
9. If $L_l(X^i_k)$ is more accurate than $L_m(X^i_k)$, $Y^i_{kml} = 1$, else $Y^i_{kml} = 0$
 end
10. For $l = 1 \rightarrow |L|$
11. For $m = l + 1 \rightarrow |L|$
12. Train an SVM, S_{ml} with $\bigcup_{i=1}^4 X^i_{ml}$ as input and $\bigcup_{i=1}^4 Y^i_{ml}$ as output
13. For $l = 1 \rightarrow |L|$ train L_l using X

end

is performed in a round-robin manner (for a loop at step 3), so all the data are used as test set once. Then, all the base learners are trained, tested, and the results are stored $L_l(X^i_k)$. Comparing the base learners based on $L_l(X^i_k)$ gives an indication of the accuracy of each of the learning algorithms for the data point X^i_k. In steps 7–9, a pairwise comparison between the learners is performed and a dataset, Y, is built that stores information about which of the learners performs better for a particular data record. In steps 10–12, an SVM is trained for each pair of learners that models where in the feature space each learner performs better. In step 13, the training data are used to train the base learners.

In the testing phase of the algorithm, in step 2, the output of all the base learners for the input data T_k is measured and stored in Z_{kl}. Then, the algorithms that are more likely to produce a more accurate result for the T_k are identified. To do so, in steps 2–6, based on the trained SVMs that model the comparison between each pair of the algorithms, the number of times each learner is predicted to perform better is counted

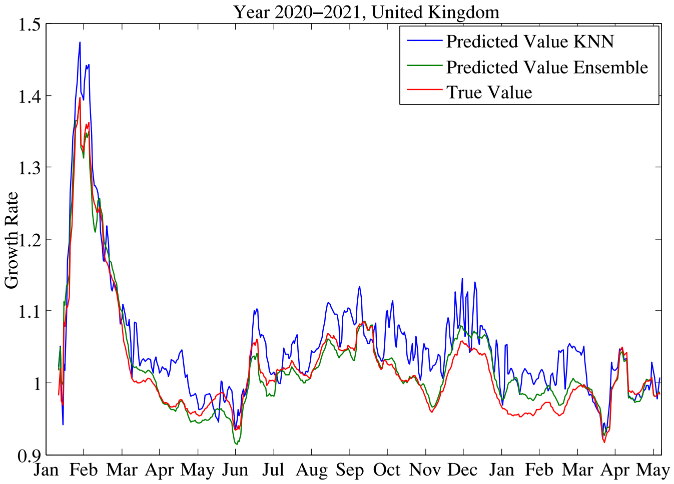

Fig. 1. Predicted and true values of growth rate of the number of cases in the U.K. using KNN and the proposed ensemble algorithm. These data are generated by using all the data except the data for the U.K. to train the models. Then, the model is tested on the U.K. data.

Algorithm 2: Testing Phase of the Proposed Ensemble Algorithm

begin
1. For $k = 1 \rightarrow |T|$
   begin
2. For $l = 1 \rightarrow |L|$ find the output of L_l on test record T_k and store it in Z_{kl}
3. For $l = 1 \rightarrow |L|$ $C_{kl} = 0$
4. For $l = 1 \rightarrow |L|$
5. For $m = l + 1 \rightarrow |L|$
6. If output of S_{ml} on T_k, $S_{ml}(T_k) = 1$, $C_{km} = C_{km} + 1$, else $C_{kl} = C_{kl} + 1$
7. Return $\frac{\sum_{l=1}^{|L|} Z_{lk} C_{lk}}{\sum_{l=1}^{|L|} C_{kl}}$ as the output for test record T_k
   end
end

and stored in C_{kl}. Then, in step 7, C_{kl} is used as a weight to measure the weighted average of the output of the base learners Z_{kl}.

By estimating the accuracy of each of the modeling algorithms via SVMs, and performing the weighted averaging, our proposed algorithm manages the uncertainty in objective evaluation.

Fig. 1 shows the growth rate of the new cases in the U.K. and its predicted value with the proposed ensemble learning and the KNN algorithms. The data for other countries are used to train the learning algorithm and the model is used to predict the values for the U.K. The data in this figure suggest that not only the proposed algorithm performs better in modeling the growth rate but also, compared to KNN, there are fewer fluctuations in the predicted value. This is true when the proposed algorithm is compared with other base learning algorithms. The reason here is that the proposed algorithm performs a weighted averaging to aggregate the output of the classifiers and this averaging reduces the uncertainty. (Note that the main approach in managing uncertainty in fitness calculation is averaging [34].)

III. Proposed Optimization Algorithm

The aim of this article is to propose a method that uses the existing data about the behavior of the pandemic and suggests the best policy to the decision makers. This process consists of two parts; one is the machine learning algorithm that models the behavior of the pandemic and the other is the optimization process that searches through different policies and suggests the policy with the lowest cost and best effect on the growth rate of infected cases. In this section, we propose the optimization process. Here, we face a multiobjective optimization problem; one criterion is the growth rate in the infection rate, and the other is the cost it induces on the society. Although the growth rate of the pandemic is straightforward, the cost each policy induces on the society is tough to measure. Therefore, in this article, we consider three scenarios for the optimization process.

The first scenario is when the policymakers have a good estimation of the cost posed by each policy to society. In this scenario, there is a two objective optimization problem of minimizing the growth rate and the cost of policies to society. In this scenario, the cost of the policies is simply calculated as

$$f = \sum_{i=1}^{|P|} g_i(x_i) \tag{1}$$

where $g_i(x_i)$ is the cost of implementing P_i as suggested by x_i. For example, as shown in Table III in the supplementary material, x_1 can take a value between $[-100, 100]$ and $g_1(x_1)$ returns the cost of implementing P_1 with the value x_1. Note that in the first scenario, it is assumed that g_i, $i = 1 \cdots |P|$ are known.

Although it is reasonable to assume that a government can make a good estimation of these costs, it is understandable that many governments may not have such measurements. To overcome this, we suggest the second scenario in which the estimated cost of each policy is not required, but the policymaker should rank the policies based on the cost they believe will inflict on society. In this case, the cost of a set of policies is calculated as

$$f = \sum_{i=1}^{|P|} \left(\frac{x_i - \min_{x_i}}{\max_{x_i} - \min_{x_i}} \right) r_i \tag{2}$$

where r_i is the rank of the policy P_i and the value of x_i is normalized between [0,1] so the weight of all policies is equal.

The third optimization scenario does not calculate an aggregate cost of the policies but considers each of the values of the policies x_i as an objective that should be minimized individually. Thus, we have a multiobjective optimization problem with $|P|+1$ objectives ($|P|$ policies plus growth rate). This scenario is devised, so a set of nondominated solutions among all policies, and the growth rate is suggested to the policy makers.

In terms of the optimization process, the first scenario is the most straightforward problem to solve; however, it requires a

Algorithm 3: Proposed Optimization Algorithm

begin
1. Set the algorithm parameters, c_1, c_2, w
2. Initialise the population X, V
3. **While** not termination condition
   begin
4. For $i = 1 \to |X|$ calculate growth rate of x_i via landscape smoothing
5. For $i = 1 \to |X|$ calculate cost of x_i to society
6. For $i = 1 \to |X|$ calculate the fitness of x_i via SPEA2
7. update gbest, pbest
8. For $i = 1 \to |X|$
   $v_i = wv_i + c_1R(0, 1)(pbest_i - x_i) + c_2R(0, 1)(gbest - x_i)$
9. For $i = 1 \to |X|$ $x_i = x_i + v_i$
10. Update the set of non dominated solutions H
   end

end

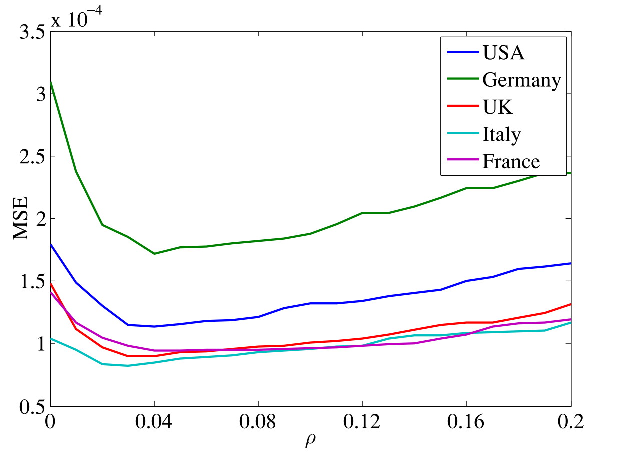

Fig. 2. Performance of the proposed ensemble learning algorithm when the averaging operator is used for different values of radius. This is averaged over all the data records for each country.

good indication of the cost that each policy causes. The second scenario is less straight forward to solve, but easier for the policymakers as there is no need of knowing the true (or estimated) cost of implementing each policy. If the policymakers do not have any indication of the cost each policy causes, the third scenario is used. In this case, because the number of objectives is large (greater than 3), most multiobjective optimization problems will not work properly. With 21 objectives, the third scenario is a "many-objective optimization problem" and specialized algorithms are required to solve the problem. Studying the ways in which the problem can be solved via this scenario remains for future work as it requires new sets of algorithm development and experiments. In terms of optimization process, clearly, this problem is harder to solve than the first and second scenarios. However, this scenario is used when the governments do not have any indication of the cost that each policy inflicts on society.

In step 4 of the Algorithm 3, the growth rate caused by the policies suggested by the particle x_i is calculated via the surrogate model. As explained before, Fig. 1 suggests that due to the averaging nature of the proposed algorithm, some of the uncertainty in calculating the fitness function is removed. However, there still remains some uncertainty in fitness calculation that affect the optimization process. Some improvements that are observed by the optimization algorithm may be due to uncertainty rather than real progress. In the literature, resampling is the main approach to managing this [34]. Resampling means that fitness is evaluated several times and a sort of averaging is performed. However, due to the deterministic nature of the modeling, resampling cannot be used in this article. To overcome this, we propose a landscape smoothing operator [35]. In the proposed algorithm, when measuring the fitness of a particular solution, instead of resampling the same solution, the fitness of the points in the landscape close to the solution is used for the averaging operator. The argument is that a small change in policies should not result in a large change in the growth rate and any sudden change in the growth rate is more likely due to uncertainty rather than a true change in the value of the growth rate. The operator we propose in this article creates a number of solutions in a radius around the current solution and performs an averaging over these solutions. Because these solutions are close to the current solution, they should have similar value in fitness and the averaging could remove the uncertainties.

Fig. 2 shows the performance of the proposed ensemble learning algorithm when the averaging operator is used for different values of the radius. Here, all the dimensions of the feature space are normalized between [0, 1] to manage the scaling problem so all the features vary between zero and one. In this method, the uncertainty is reduced by finding a number of solutions (in this case, ten solutions) on a hypersphere around the current solution at the radius ρ and find the average value as the estimated growth rate. The data in this figure are averaged over all the data records available for each country. The data in this graph suggests that at the radius around $\rho = 0.04$ (around 4% of the length of the feature space), the algorithm reaches its best performance. At $\rho = 0$, no averaging is performed. This suggests that using an averaging over the solutions in a hypersphere around the solution can remove uncertainty and improve the prediction performance of the proposed algorithm. Because the best performance is achieved at $\rho = 0.04$, in this article, we use this value for uncertainty reduction.

The second operator for reducing the uncertainty uses a characteristic of particle swarm optimization (PSO) in the search process. In PSO, because in each iteration the individuals move a small step in the search space, the change in the fitness should not be large and any sudden change should be smoothed. Fig. 3 shows the expected change in the value of growth rate due to the movement operator in PSO for a different number of steps τ. The data suggest that on average, the change in the predicted growth rate is small, for a small number of steps taken by the algorithm. This means that if the algorithm observes a sudden change in the value of the predicted growth rate, with high probability it is due to uncertainty rather than a real change due to the changes in the

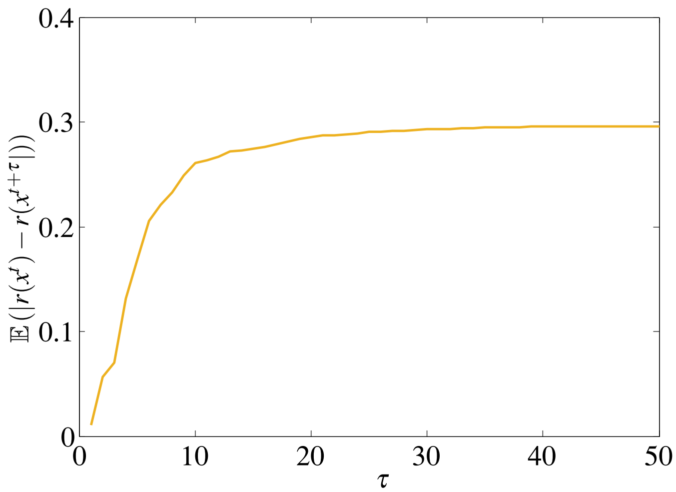

Fig. 3. Expected change in the predicted growth rate $r(.)$, as a function of the number of steps the PSO takes. This is an average over 50 runs.

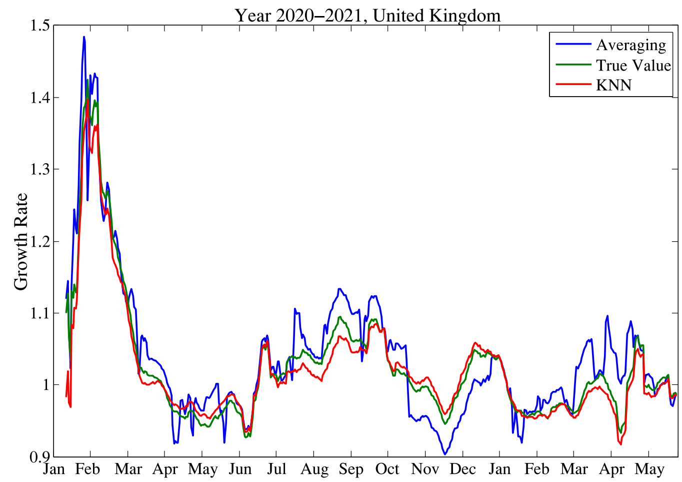

Fig. 4. Proposed landscape smoothing operator applied to KNN in predicting the growth rate for USA. Landscape smoothing operator applies both averaging and movement smoothing operators.

policies. As the algorithm takes more steps, larger changes in the predicted growth rate are observed.

Thus, the growth rate of the particle x_i^t is calculated as

$$\bar{r}(x_i^t) = \frac{1}{\sum_{j=t-\tau}^{t} e^{j-t}} \sum_{j=t-\tau}^{t} r\left(x_i^j\right) e^{j-t} \tag{3}$$

where $r(x_i^t)$ is the growth rate of the ith particle at iteration t, and $\bar{r}(x_i^t)$ is the smoothed value of the growth rate. In other words, the smoothed value of growth rate is the weighted average of the growth rate over the last τ steps the particle has taken in the search space. The weight of the growth rates in previous steps decays exponentially with the time difference, so the values of growth rate in near past have exponentially greater weight than ones in distant past. This has been devised because the more distance between two points in the search space the less correlation is expected between the fitness of the points (see Fig. 3).

In order to show how the proposed smoothing operator removes the fluctuations in the fitness landscape that are due to approximation uncertainty, Fig. 4 shows the proposed landscape smoothing operator applied to smooth the predicted value of the growth rate for the KNN algorithm. While KNN may show sudden changes in the predicted value, the proposed smoothing operator reduces the fluctuations. Removing the fluctuations has two benefits; first, it decreases the error in the prediction. Second, it removes the rapid changes in the predicted value that are due to the approximation uncertainty that can mislead the optimization algorithms. When the algorithms observe a large change in the fitness function, they are affected and change the search direction accordingly. In the case of PSO for example, when a particle observes an improve in the value of the fitness, it stores the particle in its *pbest* variable. If this improvement is due to uncertainty, then the particle will stick to this solution and will keep following this value. Thus, removing these rapid changes is crucial in the optimization process.

In step 5 of the algorithm, the cost of the set of policies x_i is calculated. The cost can be measured via any of the three scenarios mentioned before. In step 6, strength pareto evolutionary Algorithm 2 (SPEA2) [36] is used to measure the fitness of each solution. If the first or the second scenario are used, we have a two-objective optimization problem; one objective is the cost and the other is the growth rate. If the third scenario is used, we have a 21 objective-optimization problem, 20 of which correspond to the policies summarized in Table III in the supplementary material and one is the growth rate.

In step 8 of the algorithm, the PSO optimization is performed via the following equation:

$$v_i = wv_i + c_1R(0,1)\left(\text{pbest}_i - x_i\right) + c_2R(0,1)(\text{gbest} - x_i) \tag{4}$$

$$x_i = x_i + v_i \tag{5}$$

where c_1 and c_2 are positive constants, $R(.,.)$ is the uniform random number generator, w is the inertia weight, and pbest is the local and gbest is the global best solutions [37].

IV. Experimental Results

In this section, we first study the performance of the learning algorithms in modeling the pandemic based on government policies. We compare the performance of the proposed algorithm to a number of ensemble learning algorithms, including STLR [38], CBCA [38], MV [38], ABJ48 [38], RF [38], oRF10.1007, RoF [39], and MPRoF-P [39]. The performance of the learning algorithms in terms of mean square error is summarized in Table III. The data are averaged over 20 independent runs. In these experiments, the *leave-one-country-out* scheme is adopted, that is, the data for all countries except the test countries are used to train the models and the algorithms are evaluated on the test country. The best performance among the algorithms is achieved by GRNN and KNN algorithms. The worst performance is for PRN and after that RBE. Among the ensemble algorithms, the proposed SVMEL has reached the best performance among the algorithms except for Germany and France. The Friedman rank test is performed on the data and the rank of different approaches is also included in the table. For all the countries presented here, the proposed

TABLE III
MEAN SQUARE ERROR OF DIFFERENT LEARNING ALGORITHMS. THIS IS FOR WHEN THE DATA EXCEPT DATA FOR THE U.K. ARE USED FOR TRAINING AND THE DATA FOR THE U.K. ARE USED FOR TESTING. THE RESULTS ARE AVERAGED OVER 20 RUNS

		USA		GERMANY		UK		Italy		France	
	Classifier	Mean MSE	Rank	Mean MSE	Rank	Mean MSE	Rank	Mean MSE	Rank	Mean MSE	Rank
Base Learners	RBN	5.552e-03	13.5	3.492e-03	15	2.786e-03	13.7	2.123e-03	10.4	2.418e-03	14
	LVQ	6.397e-03	15.1	3.925e-03	16.5	3.321e-03	15.3	2.316e-03	12	2.821e-03	15.8
	PNN	6.410e-03	15.1	3.922e-03	16.5	3.321e-03	15.3	2.312e-03	12	2.831e-03	15.8
	RBE	6.465e-03	16.6	3.942e-03	18	3.342e-03	16.8	2.337e-03	13.5	2.855e-03	17.4
	CFNN	4.493e-03	14.2	1.825e-03	8.9	3.676e-03	13.9	2.462e-03	15.4	1.624e-03	7.7
	PRN	6.700e-04	18.2	6.899e-04	19	8.774e-04	19	7.538e-04	18.5	7.231e-04	19
	FFNN	1.128e-02	12.6	1.577e-03	7.5	1.737e-03	11	5.598e-03	16.2	1.908e-03	9.4
	GRNN	2.106e-03	10.7	1.669e-03	7.3	1.870e-03	11.1	2.642e-03	14.7	1.413e-03	6
	KNN	2.306e-03	11.7	1.731e-03	8.5	2.023e-03	12.3	2.751e-03	15.8	1.475e-03	8.3
	FNN	1.355e-02	16.6	1.984e-03	9.9	4.638e-03	16.3	3.978e-03	16.5	2.985e-03	13.1
Ensembles	SVMEL	3.161e-04	1	9.625e-04	2.1	1.808e-04	1	9.892e-04	1	1.478e-04	2.1
	STLR	8.397e-04	7.2	6.543e-04	1	4.945e-04	7.1	1.928e-03	8	4.890e-04	10.8
	CBCA	9.600e-04	9.2	1.514e-03	5.8	3.380e-04	4	1.463e-03	3	2.533e-04	7.2
	MV	7.512e-04	6.1	2.054e-03	12.3	3.760e-04	5	2.242e-03	6	3.722e-04	12.9
	ABJ48	5.368e-04	4	1.052e-03	3.2	2.379e-04	2	7.304e-04	2	2.238e-04	1
	RF	8.821e-04	8.2	2.250e-03	13.5	5.489e-04	8.1	1.648e-03	7	4.152e-04	9.6
	oRF	4.647e-04	3	1.766e-03	9.5	4.341e-04	6	1.089e-03	4	3.027e-04	3.2
	RoF	6.470e-04	5	1.337e-03	4.6	5.966e-04	9.1	1.344e-03	9	5.230e-04	4.8
	MPRoF-P	4.070e-04	2	1.865e-03	10.9	2.668e-04	3	2.118e-03	5	3.215e-04	11.9

algorithm has reached the best rank for all the experiments. The same experiments are performed on the data from India and Brazil as the mostly affected countries, China, as the country from which the pandemic started and New Zealand as a country with very effective policies. The data are presented in Table I in the supplementary material.

In order to statistically test the proposed algorithm, Kruskal–Wallis and two-tailed Wilcoxon tests are performed in this article and the data are presented in Table IV in the supplementary material. In this table, "SS" is the sum of squares of each source, "df" is the degree of freedom associated with each source, "MS" is the mean squares (the ratio SS/df), and "Chi-square" is the ratio of mean squares. The p-values represent the probability that the samples are taken from populations with the same means. The small p-values in this table are small, which indicates that the null hypothesis that all the samples are taken from the same mean is rejected.

Fig. 3 in the supplementary material presents the box plot of the results of different algorithms for different countries. A comparison between the base learners with the ensemble learning algorithms suggests that not only the ensemble algorithms reach better performance but also, with a smaller standard deviation, the results are more consistent. This is the case for the experiments on France, USA, and United Kingdom. For Germany, the standard deviations for the base learners and the ensembles have similar values.

The predictions offer good results, suggesting the data from the countries around the world can be used to predict the behavior of the pandemic in another country, which means that the response of the pandemic behavior to policies is very similar in different countries.

In order to compare the performance of the proposed SVMEL with the base learning algorithms and other ensemble algorithms, Table IV shows the performance of different learning algorithms averaged over all the countries. To generate these data, the algorithms are tested on all the countries and the average MSEs are reported. For each country, the training

TABLE IV
PERFORMANCE OF DIFFERENT ALGORITHMS AVERAGED OVER ALL THE COUNTRIES. THE DATA FOR ALL COUNTRIES EXCEPT THE TEST COUNTRY ARE USED TO TRAIN THE LEARNING ALGORITHMS. THE COLUMN RANK SHOWS THE FRIEDMAN RANK OF THE ALGORITHMS. MEAN REPRESENTS THE AVERAGE AND STD REPRESENTS THE STANDARD DEVIATION OF THE RESULTS

	Classifier	Mean MSE	STD	Rank
Base Learners	RBN	4.383e-03	3.252e-03	12.4
	LVQ	4.786e-03	3.459e-03	13.9
	PNN	4.797e-03	3.467e-03	13.9
	RBE	4.204e-03	3.483e-03	14.2
	CFNN	1.201e-02	5.229e-02	12.6
	PRN	7.720e-02	9.979e-02	18.9
	FFNN	1.187e-02	8.460e-02	12.1
	GRNN	3.689e-03	3.563e-03	11
	KNN	3.803e-03	3.594e-03	11.9
	FNN	2.367e-02	1.364e-01	14.5
Ensembles	SVMEL	7.538e-04	1.818e-10	1.3
	STLR	1.529e-03	9.271e-10	6
	CBCA	2.031e-03	7.140e-10	9.9
	MV	1.624e-03	8.324e-10	7.2
	ABJ48	1.246e-03	4.965e-10	4.4
	RF	2.366e-03	9.632e-10	12.2
	oRF	1.899e-03	8.433e-10	8.8
	RoF	8.710e-04	1.139e-10	2.3
	MPRoF-P	1.043e-03	9.362e-10	3.3

data are created to contain the data for all the countries except the testing country. The last column shows the Friedman rank test. Among the algorithms, the proposed algorithm reaches rank first, followed by RoF and MPRoF-P.

Fig. 5 shows the Pareto front for the first scenario found by the proposed optimization algorithm when the proposed ensemble method is used for predicting the growth rate. Because we did not have access to the data estimating the cost of each policy to the society, in this experiment, we set an example of costs in Table V. Note that the costs are different in different countries, so the policy makers should use their own cost table.

In this article, we compare a number of optimization algorithms. The parameters of the optimization algorithms are set

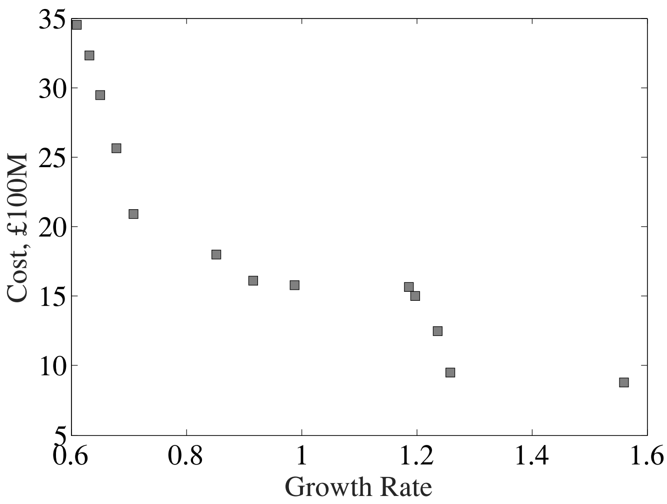

Fig. 5. Pareto front of the policies achieved by the proposed algorithm. The costs are in £100M and the cost of each policy is assumed as Table V.

TABLE V
EXAMPLE OF THE COST OF EACH POLICY INFLICTED TO 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	8	1	9	4	6	6	5	9	9	7

Policy	P_{11}	P_{12}	P_{13}	P_{14}	P_{15}	P_{16}	P_{17}	P_{18}	P_{19}	P_{20}
Cost	8	4	4	3	5	6	3	5	7	9

TABLE VI
MEAN HV OF THE PARETO-FRONT FOR THE FIRST SCENARIO OPTIMIZATION WITH THE COSTS PRESENTED IN TABLE V. MEAN REPRESENTS THE AVERAGE AND STD REPRESENTS THE STANDARD DEVIATION OF THE RESULTS OVER 30 INDEPENDENT RUNS

Optimiser	Mean MSE	STD	Rank
PSO+Smoothing	34.64	1.02	1.5
PSO	33.83	1.24	2.2
GA	29.75	0.70	9.2
DE	32.34	0.73	4.3
ES	30.21	1.12	8.4
FES	32.92	0.33	3.1
EP	31.6	0.41	5.7
FEP	28.14	1.02	11.7
MASW	30.75	0.56	7.4
EDA	31.4	0.96	6
RCODE	29.88	0.71	8.9
LLSO	29.17	1.22	10.2

as GA [40] (mutation rate = 0.05 and crossover rate 0.7), PSO [41] ($c_1 = 2$, $c_2 = 2$, and $w = 0.1$), evolutionary programming (EP, parameters set as [42]), fast evolutionary programming (FEP, parameters set as [43]), evolutionary strategy (ES, parameters set as [44]), fast ES (FES, $L = 1$ and $S = 1.1$) [45], Ma-ssw-chains (MASSW, parameters set as [46]), and differential evolution (DE, $F = 0.2$ and $O = 0.8$). In order to compare the performance of different optimization algorithms, Table VI shows the mean hypervolume (HV) of the Pareto-front for the first scenario optimization with the cost of each policy presented in Table V. The Friedman rank test suggests that PSO achieves the best performance among the optimization algorithms and its performance is improved when the landscape smoothing operator in (3) is used. For all the algorithms, we use the averaging operator with $\rho = 0.04$ to reduce the uncertainty with 10 samplings. That is, when

TABLE VII
SOME OF THE SOLUTIONS IN THE PARETO-FRONT WHEN THE THIRD SCENARIO OPTIMIZATION IS PERFORMED

Rate	1.06	0.83	0.84	0.81	1.42	0.93	0.87	0.84
P_1	66	6	87	-18	73	41	-19.39	46.59
P_2	-87	60	70	42	78	66.16	-43.36	13.74
P_3	-77	61	88	53	32	74.82	-100	-51.18
P_4	15	50	49	52	29	14.36	14.28	18.31
P_5	3	3	0	0	2	0.92	0	2.43
P_6	1	0	1	0	0	1	0	0.67
P_7	67	16	93	29	29	76.36	0	42.71
P_8	1	1	0	2	2	2	0.33	1.93
P_9	2	1	0	1	2	1.03	2	1.97
P_{10}	0	1	1	0	1	0.17	0	1
P_{11}	0	0	4	2	0	1.09	4	2
P_{12}	2	0	1	0	1	2	2	1.24
P_{13}	0	1	1	1	1	2	0	0
P_{14}	1	4	1	4	4	0.23	2.24	4
P_{15}	-15	-35	-89	-91	40	-100	33.83	3.59
P_{16}	3	2	3	1	3	1.24	1.27	2
P_{17}	1	2	1	1	1	2.1	2.96	0.84
P_{18}	2	0	2	2	0	0.4	0	1.05
P_{19}	-29	97	-84	12	9	-79.64	-100	48.42
P_{20}	3	1	2	1	3	0	0	3

estimating the growth rate for a solution, ten solutions at a radius $\rho = 0.04$ around the solutions are generated randomly and an averaging is performed.

For the third scenario, the algorithm returns the Pareto-front where each policy is considered as an objective. Table VII shows eight of the solutions in the Pareto-front. These are nondominated solutions and the policy makers should choose a policy from this list. When the estimated cost of each policy to the society is unknown, or for some cases like closing the schools is hard to estimate, the policy makers should choose from these values subjectively. For example, if the goal is to minimize the growth rate at any cost without closing schools, a solution with minimum growth rate that suggests opening school can be used.

V. CONCLUSION

In this article, we proposed an evolutionary algorithm with a surrogate ensemble learning algorithm that performs as a fitness function to find the optimal government policy against the spread of the virus. We used data about the policies taken by 183 countries and the data about the number of cases in these countries to build a model that takes as input the policies taken by a country and generates as output the growth rate of the infected cases. To build this model, we propose an ensemble machine learning algorithm that consists of ten base learning algorithms. In the proposed ensemble, in order to aggregate the output of the base learning algorithms, SVM is used to predict the performance of each of the algorithms. The experiments performed in this article suggest an improved performance of the proposed algorithm. We then use this model as a fitness function to build a multiobjective optimization algorithm.

The proposed algorithm was trained on the data from the beginning of the pandemic, until the 20th of August, the date of writing of this article. As time progresses and more data are available, more accurate models can be created and thus, more reliable policies are suggested by the proposed algorithm. Also, as more policies are devised by the governments, and more data are available, the accuracy of the models and the suggested policies will improve. At the moment, one complication is data availability and accuracy. Not all governments release data about the policies they take. Governments should provide more organized information about their policies and cases. Also, there is noise in the data. The number of cases reported by governments almost never presents the true number of infected cases. One is because of different testing policies between governments. The other is transparency as some countries may not report the true number of cases. Unless these obstacles are removed, the modeling algorithms are destined to suffer from inaccuracy.

Because of the nature of modeling algorithms, the optimization algorithms should adopt an uncertainty reduction mechanism. In the literature, averaging is usually used to reduce noise. The uncertainty in modeling problems is the approximation uncertainty [1], [47] and because it is deterministic, it cannot be removed via averaging. In this article, we proposed two uncertainty reduction schemes. The first scheme is to find a number of solutions in a radius around the current solution and perform an average over the estimated growth rate of these solutions. We showed this method not only reduces the fluctuations due to the uncertainty but also improves the prediction accuracy. Also, the uncertainty may cause fluctuations in the fitness landscape that should be removed. We proposed a movement smoothing operator that reduces sudden changes in the fitness that are more likely due to noise. Another field of research is to study the fitness landscape properties of the optimization problem [48]–[50].

In the future, there could be any changes to the situation, for example, the discovery of vaccines. These changes would not change the applicability of the algorithm. If a vaccine is discovered, the proposed algorithm can still be used with vaccination being another policy added to the set of policies. Up to the point of the publication of this article, not many countries have started vaccination and the ones who have implemented the policy have not vaccinated enough of the population to clearly benefit from the effects. One line of future work can be the study of the effect of vaccination on the pandemic. There are many types of vaccines developed at the moment. These vaccines are different in their prices, cost of vaccination, effectiveness, etc. The proposed approach can be generalized to study the vaccination policies and to find the best policies in that regard.

At the time of the publication of this article, a number of variations of the virus have evolved in some countries, including the 20B/501Y.V1 variant (colloquially knows as U.K. variant) or 501Y.V2 variant (known as South Africa variant). Because these variants are known to have a higher infection rate, the behavior of the pandemic relating to these variants will be different. Also, there are many factors that change the behavior of the pandemic, including the level of herd immunity, the appearance of new variants, etc. The current version of the work can be improved in future work to consider these challenges. One approach could be to employ the algorithms that detect and manage concept drift. This remains for future work.

This pandemic has not been the first and will not be the last one that has affected humanity's life [51]. In the current patterns in the relationship between humans and wildlife, it is expected to have an increasing number of Zoonotic disease pandemics in the future [52]. The current pandemic may be over soon, but the need for approaches to tackle future pandemics remains. This article should be considered as a stepping stone for the way AI approaches can be used to tackle pandemics.

References

[1] Y. Jin and J. Branke, "Evolutionary optimization in uncertain environments–A survey," *IEEE Trans. Evol. Comput.*, vol. 9, no. 3, pp. 303–317, Jun. 2005.

[2] J. A. Benediktsson, J. R. Sveinsson, O. K. Ersoy, and P. H. Swain, "Parallel consensual neural networks," *IEEE Trans. Neural Netw.*, vol. 8, no. 1, pp. 54–64, Jan. 1997.

[3] L. Breiman, "Stacked regressions," *Mach. Learn.*, vol. 24, no. 1, pp. 49–64, 1996.

[4] L. K. Hansen and P. Salamon, "Neural network ensembles," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 10, pp. 993–1001, Oct. 1990.

[5] S. Hashem, "Optimal linear combinations of neural networks," *Neural Netw.*, vol. 10, no. 4, pp. 599–614, 1997.

[6] L. I. Kuncheva, "Diversity in multiple classifier systems," *Inf. Fusion*, vol. 6, no. 1, pp. 3–4, 2005.

[7] R. Polikar, "Ensemble based systems in decision making," *IEEE Circuits Syst. Mag.*, vol. 6, no. 3, pp. 21–45, 3rd Quart., 2006.

[8] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.

[9] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 832–844, Aug. 1998.

[10] R. Ranawana and V. Palade, "Multi-classifier systems: Review and a roadmap for developers," *Int. J. Hybrid Intell. Syst.*, vol. 3, no. 1, pp. 35–61, 2006.

[11] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, 1997.

[12] P. Bühlmann and B. Yu, "Analyzing bagging," *Ann. Stat.*, vol. 30, no. 4, pp. 927–961, 2002.

[13] A. Buja and W. Stuetzle, "Observations on bagging," *Statistica Sinica*, vol. 16, no. 2, pp. 323–351, 2006.

[14] L. Breiman, "Pasting small votes for classification in large databases and on-line," *Mach. Learn.*, vol. 36, nos. 1–2, pp. 85–103, 1999.

[15] R. Bryll, R. Gutierrez-Osuna, and F. Quek, "Attribute bagging: Improving accuracy of classifier ensembles by using random feature subsets," *Pattern Recognit.*, vol. 36, no. 6, pp. 1291–1302, 2003.

[16] J. Stefanowski, "Combining answers of sub-classifiers in the bagging-feature ensembles," in *Proc. Int. Conf. Rough Sets Intell. Syst. Paradigms*, 2007, pp. 574–583.

[17] Q.-T. Cai, C.-Y. Peng, and C.-S. Zhang, "A weighted subspace approach for improving bagging performance," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2008, pp. 3341–3344.

[18] T. Zhou, H. Lu, Z. Yang, S. Qiu, B. Huo, and Y. Dong, "The ensemble deep learning model for novel COVID-19 on CT images," *Appl. Soft Comput.*, vol. 98, Art. no. 106885, Jan. 2021.

[19] A. K. Das, S. Ghosh, S. Thunder, R. Dutta, S. Agarwal, and A. Chakrabarti, "Automatic COVID-19 detection from X-ray images using ensemble learning with convolutional neural network," *Pattern Anal. Appl.*, vol. 24, no. 2, pp. 1–14, 2021.

[20] M. AlJame, I. Ahmad, A. Imtiaz, and A. Mohammed, "Ensemble learning model for diagnosing COVID-19 from routine blood tests," *Informat. Med. Unlocked*, vol. 21, Jan. 2020, Art. no. 100449.

[21] N. E. Dean *et al.*, "Ensemble forecast modeling for the design of covid-19 vaccine efficacy trials," *Vaccine*, vol. 38, no. 46, pp. 7213–7216, 2020.

[22] S. Benítez-Peña *et al.*, "On sparse ensemble methods: An application to short-term predictions of the evolution of COVID-19," *Eur. J. Oper. Res.*, vol. 295, no. 2, pp. 648–663, 2021.

[23] N. Hasan, "A methodological approach for predicting COVID-19 epidemic using EEMD-ANN hybrid model," *Internet Things*, vol. 11, Sep. 2020, Art. no. 100228.

[24] A. I. Saba and A. H. Elsheikh, "Forecasting the prevalence of COVID-19 outbreak in egypt using nonlinear autoregressive artificial neural networks," *Process Safety Environ. Protect.*, vol. 141, pp. 1–8, Sep. 2020.

[25] P. Melin, J. C. Monica, D. Sanchez, and O. Castillo, "Multiple ensemble neural network models with fuzzy response aggregation for predicting COVID-19 time series: The case of mexico," *Healthcare*, vol. 8, no. 2, p. 181, 2020.

[26] P. Hartono, "Similarity maps and pairwise predictions for transmission dynamics of COVID-19 with neural networks," *Informat. Med. Unlocked*, vol. 20, Jan. 2020, Art. no. 100386.

[27] R. S. Pontoh, T. Toharudin, S. Zahroh, and E. Supartini, "Effectiveness of the public health measures to prevent the spread of COVID-19," *Commun. Math. Biol. Neurosci.*, vol. 2020, p. 31, Jun. 2020.

[28] P. Wang, X. Zheng, J. Li, and B. Zhu, "Prediction of epidemic trends in COVID-19 with logistic model and machine learning technics," *Chaos Solitons Fractals*, vol. 139, Oct. 2020, Art. no. 110058.

[29] L. A. Amar, A. A. Taha, and M. Y. Mohamed, "Prediction of the final size for COVID-19 epidemic using machine learning: A case study of egypt," *Infect. Disease Model.*, vol. 5, pp. 622–634, 2020.

[30] S. Tuli, S. Tuli, R. Tuli, and S. S. Gill, "Predicting the growth and trend of COVID-19 pandemic using machine learning and cloud computing," *Internet Things*, vol. 11, Sep. 2020, Art. no. 100222.

[31] K. Raza, *Artificial Intelligence Against COVID-19: A Meta-Analysis of Current Research*. Cham, Switzerland: Springer, 2020, pp. 165–176.

[32] C. M. Yeşilkanat, "Spatio-temporal estimation of the daily cases of COVID-19 in worldwide using random forest machine learning algorithm," *Chaos Solitons Fractals*, vol. 140, Nov. 2020, Art. no. 110210.

[33] M.-H. Tayarani N., "Applications of artificial intelligence in battling against COVID-19: A literature review," *Chaos Solitons Fractals*, vol. 142, Jan. 2020, Art. no. 110338.

[34] P. Rakshit, A. Konar, and S. Das, "Noisy evolutionary optimization algorithms–A comprehensive survey," *Swarm Evol. Comput.*, vol. 33, pp. 18–45, Apr. 2017.

[35] M. Tayarani-N. and A. Prügel-Bennett, "On the landscape of combinatorial optimization problems," *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 420–434, Jun. 2014.

[36] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm," Eidgenössische Technische Hochschule Zürich (ETH), Institut für Technische Informatik und Kommunikationsnetze (TIK), Rep. 103, 2001.

[37] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. ICNN Conf. Neural Netw.*, vol. 4, 1995, pp. 1942–1948.

[38] A. Jurek, Y. Bi, S. Wu, and C. D. Nugent, "Clustering-based ensembles as an alternative to stacking," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 9, pp. 2120–2137, Sep. 2014.

[39] L. Zhang and P. N. Suganthan, "Oblique decision tree ensemble via multisurface proximal support vector machine," *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2165–2176, Oct. 2015.

[40] J. H. Holland, *Adaption in Natural and Artificial Systems*, 1st ed. Ann Arbor, MI, USA, MIT Press, 1975.

[41] J. Kennedy, R. C. Eberhart, and Y. H. Shi, *Swarm Intelligence* (The Morgan Kaufmann Series in Evolutionary Computation), 1st ed. Burlington, MA, USA: Morgan Kaufmann, 2001.

[42] L. J. Fogel, A. J. Owens, and M. J. Walsh, *Artificial Intelligence Through Simulated Evolution*. New York, NY, USA: Wiley, 1966.

[43] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Trans. Evol. Comput.*, vol. 3, no. 2, pp. 82–102, Jul. 1999.

[44] H.-P. Schwefel, *Evolution and Optimum Seeking*. New York, NY, USA: Wiley, 1995.

[45] X. Yao and Y. Liu, "Fast evolution strategies," *Control Cybern.*, vol. 26, pp. 467–496, Apr. 1997.

[46] D. Molina, M. Lozano, A. Sánchez, and F. Herrera, "Memetic algorithms based on local search chains for large scale continuous optimisation problems: MA-SSW-chains," *Soft Comput.*, vol. 15, no. 11, pp. 2201–2220, 2011.

[47] M.-H. Tayarani-N., X. Yao, and H. Xu, "Meta-heuristic algorithms in car engine design: A literature survey," *IEEE Trans. Evol. Comput.*, vol. 19, no. 5, pp. 609–629, Oct. 2015.

[48] M.-H. Tayarani-N. and A. Prügel-Bennett, "An analysis of the fitness landscape of travelling salesman problem," *Evol. Comput.*, vol. 24, no. 2, pp. 347–384, 2016.

[49] M.-H. Tayarani-N. and A. Prügel-Bennett, "Quadratic assignment problem: A landscape analysis," *Evol. Intell.*, vol. 8, no. 4, pp. 165–184, 2015.

[50] M.-H. Tayarani-N. and A. Prügel-Bennett, "Anatomy of the fitness landscape for dense graph-colouring problem," *Swarm Evol. Comput.*, vol. 22, pp. 47–65, Jun. 2015.

[51] World Health Organization, Geneva, Switzerland, "Prioritizing Diseases For Research and Development in Emergency Contexts," 2020. [Online]. Available: https://www.who.int/activities/prioritizing-diseases-for-research-and-development-in-emergency-contexts

[52] H. S. Young *et al.*, "Declines in large wildlife increase landscape-level prevalence of rodent-borne disease in Africa," vol. 111, no. 19, pp. 7036–7041, 2014.