BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **23**, No 1
Sofia • 2023 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2023-0007

Improved Barnacle Mating Optimizer-Based Least Square Support Vector Machine to Predict COVID-19 Confirmed Cases with Total Vaccination

Marzia Ahmed[1,2], *Mohd Herwan Sulaiman*[1], *Ahmad Johari Mohamad*[1]

[1]*Faculty of Electrical & Electronics Engineering Technology, University Malaysia Pahang, Pekan, Malaysia*
[2]*Department of Software Engineering, Daffodil International university, Daffodil Smart City, Ashulia, Dhaka, Bangladesh*
E-mails: ahmed.marzia32@gmail.com herwan@ump.edu.my johari@ump.edu.my

Abstract**:** *Every country must have an accurate and efficient forecasting model to avoid and manage the epidemic. This paper suggests an upgrade to one of the evolutionary algorithms inspired by nature, the Barnacle Mating Optimizer* (*BMO*)*. First, the exploration phase of the original BMO is enhanced by enforcing and replacing the sperm cast equation through Levy flight. Then, the Least Square Support Vector Machine* (*LSSVM*) *is partnered with the improved BMO* (*IBMO*)*. This hybrid approach, IBMO-LSSVM, has been deployed effectively for time-series forecasting to enhance the RBF kernel-based LSSVM model since vaccination started against COVID-19 in Malaysia. In comparison to other well-known algorithms, our outcomes are superior. In addition, the IBMO is assessed on 19 conventional benchmarks and the IEEE Congress of Evolutionary Computation Benchmark Test Functions* (*CECC06, 2019 Competition*)*. In most cases, IBMO outputs are better than comparison algorithms. However, in other circumstances, the outcomes are comparable.*

Keywords**:** *COVID-19 confirmed case, Total vaccinations, Barnacles mating optimizer, Levy flight, Meta-heuristic, Least Square Support Vector Machine* (*LSSVM*)*.*

1. Introduction

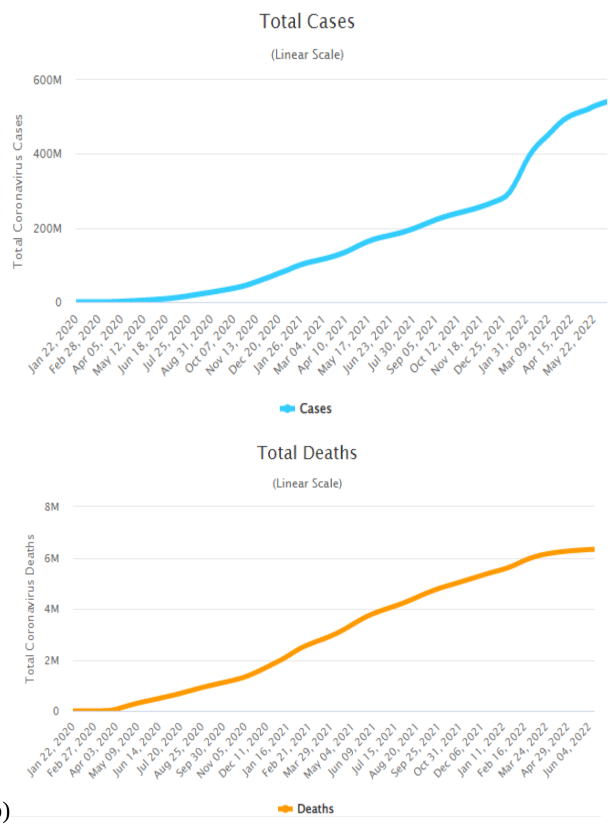
Wuhan, China, was the site of the first official discovery of COVID-19, a novel coronavirus. The virus has now claimed millions of lives throughout the globe. Interstitial pneumonia and Acute Respiratory Distress Syndrome (ARDS) are the most common symptoms of COVID-19, an infectious illness caused by SARS-CoV-2. COVID-19's acute phase is generally handled by infectious disease doctors, neurologists, and critical care physicians [1]. COVID-19 cases have reached more than five million worldwide, yet the proportion of patients who have survived is increasing. SARS-CoV-2 is causing much harm, and doctors and pathologists are attempting to figure out where what and how much of it is due to the virus [2]. According to the Centers for Disease Control and Prevention, close social contact, contacting surfaces, or things containing viral particles are among the ways the virus may be transferred. COVID-19 has an incubation period of up to 14 days, according to [3], and it may spread to other people during this time. In addition, a Chinese team reports in [4] that the median incubation duration is 3 days, ranging from 0 to 24 days and that the average life expectancy is 47.0 years. There has been a steady rise in confirmed cases, particularly in China. Many Chinese cities, particularly in Hubei province, have already developed stricter laws and procedures in response to the threat posed by the spread of this virus. As a result, it is essential to know how many confirmed cases there will be in the next several days to prepare accordingly. Analysis of weekly COVID-19 epidemic trends will be examined in this research.

Table1. Published hybrid time-series forecasting methods with their performance metrics

SL#	Time series forecasting methods (Confirmed cases COVID-19)	Performance matric	Refe-rence
1	Gradient-based optimizer Variational Mode Decomposition (GVMD), Extreme Learning Machine (ELM), and AutoRegressive Integrated Moving Average (ARIMA), named GVMD-ELM-ARIMA	MAPE = 9.55×10^{-5} RMS = 1.51×10^{3} MAE= 9.92×10^{2}	[17]
2	BMO-LSSVM, CS-LSSVM, GWO-LSSVM, MFO-LSSVM	MAPE (%) = 0.282, 0.349, 0.416, 0.353	[15]
3	Two-piece scale mixture normal distributions, called TP-SMN-AR models	Accuracy 98%	[18]
4	MLP	RMSE = 180.2759, MAE = 142.8951, MAPE = 5.72562	[19]
5	Long Short-Term Memory (LSTM) network	RMSE error is about 45.70 with an accuracy of 92.67%	[20]
6	Different Deep Learning approaches, outperforms: Variational AutoEncoder (VAE)	RMSE = 4,079,244, MAE = 3,976,682, MAPE = 2,04	[21]
7	Convolution-LSTM	Accuracy =97.82 (India), and 98 (USA)	[22]
8	ARIMA and Prophet time series forecasting	India confirmed cases: MAPE = 16.72 and 21.43, MAE = 7007.09 and 10245.17, RMSE = 7330.03 and 12085.37	[23]
9	LSTM, RNN (Dataset of Malaysia, Morocco and Saudi Arabia)	Accuracy: 98.58%, 93.45%	[24]

To this day, different methodologies and approaches, ranging from the more traditional ones to the more cutting-edge metaheuristic ones, have been presented in various pieces of published research. Conventional approaches include Successive Quadratic Programming (SQP) [5-7], Linear Programming (LP), and Mixed-Integer Linear Programming (MILP) [8-11] are some of the ways that have been suggested. Since traditional approaches perform poorly in terms of robustness and accuracy, it is impossible to avoid including metaheuristic methods when calculating the weekly average number of Covid-19 confirmed cases. Additionally, the regression time series model is a versatile technique that may be used to model dependent data. It has been used to estimate and predict real-world practical issues; for more information, see the references listed below [12-14]. Even if metaheuristic approaches can solve many difficult optimization issues, there is no assurance that the solutions will be optimum overall [15, 16].

Academics have used several hybrid models; Table 1 summarizes the models that have been published and includes metrics for assessing their success when used for time-series forecasting.



(a)

(b)

Fig. 1. Total confirmed cases (a) and deaths (b) from the beginning to the present date [29]

Numerous metaheuristic algorithms have been implemented in solving different kinds of optimization problems, such as Particle Swarm Optimization (PSO) [25], Gravitational Search Algorithm (GSA) [26], Ant-Lion Optimizer (ALO), Grey Wolf Optimizer (GWO) [27], Whale Optimization Algorithm (WOA), Moth-Flame Optimization Algorithm (MFO Algorithm) [28] and many more.

Fig. 1 depicts the number of confirmed cases (a) of COVID-19 that have been reported around the globe from the beginning of the outbreak to the present day. Also included is the total number of fatalities (b) that have resulted from these cases. It is not appropriate for a single model to accurately predict the number of cases in a pandemic. In order to effectively manage any form of risk, it is a must to apply newly developed and hybridized approaches that are more accurate.

The contributions of this research work can be outlined as follows:

1. A novel hybrid model, IBMO-LSSVM presented here, can predict COVID-19 confirmed cases more accurately when combined with the total number of vaccination cases, leading to better overall performance.
2. The original Barnacle Mating Optimizer (BMO), improved by Levy Flight and combined with the Least Square Support Vector Machine, improves time-series forecasting for infectious disease scenario analysis.
3. Classical benchmark and CEC2019 functions have been used to compare the performance of BMO versions and other hybrid state-of-the-art models.

This paper is structured as follows: Section 2 briefly discusses the development of the proposed improved BMO. Section 3 represents the LSSVM model in short, and Section 4 explains the dataset modelling, input-output modelling, and hybridization of IBMO-LSSVM. Finally, results are dedicated to Sections 5, and the conclusion of this paper is drawn in Section 6.

2. Improved barnacle mating optimizer

The Barnacles Mating Optimizer (BMO) is a bio-inspired algorithm based on the barnacle's mating behavior that has been proposed in [30-32]. In nature, barnacles' mating happens by the normal technique of copulation and a sperm-casting technique. In BMO, the Hardy-Weinberg principle's Punnet square concept is used in normal copulation and is treated as an exploitation process, whereas the sperm cast is treated as an exploration process. The improvement of the original BMO is in the **exploration process,** which will be presented in the following sub-topic.

2.1. Initialization

For initialization, the candidate of solution X is the population of barnacles, which can be represented as follows [30]:

(1) $$X=\begin{bmatrix} x_1^1 & \cdots & x_1^N \\ \vdots & \ddots & \vdots \\ x_n^1 & \cdots & x_n^N \end{bmatrix},$$

where n and N are the total populations and the number of control variables to be optimized, each population is evaluated, and the best solution is sorted to the top of the population.

2.2. The process of selection for parents to be mated

The process of selecting two parents of barnacles is based on the pre-determined parameter, which is reflected in the length of the barnacle's penis, namely pl. This paper will use all the assumptions discussed in [30].

2.3. Barnacles' off-spring reproduction process

The process of reproduction of BMO can be found in the following expressions:

(2) $$x_i^{N\_\text{new}} = px_{\text{barnacle}\_d}^N + qx_{\text{barnacle\_m}}^N,$$

(3) $$x_i^{N\_\text{new}} = \text{rand}(\ \ ) \times x_{\text{barnacle\_m}}^N,$$

$x^N_{\text{barnacle\_d}}$ and $x^N_{\text{barnacle\_m}}$ are the control variables of barnacles' parents, p is the normally distributed random number, and $q = 1 - p$. In this algorithm, rand() is the simple random number [0-1].

2.4. Implementation of Levy flights

In BMO, the value of pl is crucial in defining the exploration and exploitation processes. From Equation (3), it can be noted that the simple random numbers to generate new offspring are treated as an exploration process. In this paper, the improvement to the exploration process is made where Equation (3) is changed to the following expression:

(4) $$x_i^{N\_\text{new}} = x_{\text{barnacle\_m}}^n + \text{Levy}(N),$$

where Levy flight is determined as follows:

(5) $$\text{Levy}(N) = 0.01 \times \frac{r_1 \times \sigma}{|r_2|^{\frac{1}{\beta}}},$$

β is a constant set to 1.5, r_1 and r_2 are random numbers [0-1], and the equation is as follows:

(6) $$\sigma = \left(\frac{\tau(1+\beta) \times \sin\left(\frac{\pi\beta}{2}\right)}{\tau\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}} \right)^{\frac{1}{\beta}},$$

where $\tau(y) = (y-1)!$

3. Least Squares Support Vector Machines (LSSVM)

In high-dimensional feature space, SVM linear regression is performed through nonlinear mapping. It necessitates the solution of a large-scale quadratic programming problem, which is one of the SVM's major flaws. Consequently, the Least-Squares Support Vector Machine (LSSVM) has been created to handle optimization problems using linear equations rather than quadratic programming [21-23]. The LSSVM regression model can be expressed as follows:

(7) $$y(x) = \sum_{i=1}^{n} \alpha_i k(x_i, x_j) + b,$$

where, $k(x_i, x_j)$ denoted as kernel function such as linear, polynomial and multilayer perceptron, Radial Basis Function (RBF). α_i, x_i, b represents the Lagrange multipliers, i-th support vector and bias parameter, respectively.

Compared to other kernel RBF has proven its superiority in providing outstanding performance. RBF kernel in this study has been defined as follows:

(8) $$k(x_i, x_j) = \exp\left(\frac{|x_i - x_j|^2}{2\sigma^2}\right).$$

4. Proposed methodology

This section explains how the research has been done. This project has four main components: data collection, experiment setup, BMO-Levy-LSSVM architecture, and performance assessment.

4.1. BMO-Levy-LSSVM

The LSSVM hyper-parameters, γ and σ_2, are tuned in this research using the Improved BMO through Levy Flights. To forecast future instances of COVID-19, LSSVM uses the IBMO's predicted optimal values. It is important to keep in mind that LSSVM's prediction performance is very sensitive to the settings of its hyperparameters. Because of this, IBMO has to be capable of generating best-case scenarios. The objective function of this research is based on the Mean Absolute Percentage Error (MAPE), Theils'U and accuracy. The hybridization procedure aims to identify the hyper-parameter values that provide the most precise forecasts. Pseudo code for the hybrid IBMO-LSSVM is shown in Fig. 3. In line 11, the algorithm has been refined using Levy Flights, and in lines 3 and 14, the LSSVM method has been incorporated to assess the fitness function via training and validation. The process flow of IBMO implementation is exhibited in Fig. 2.

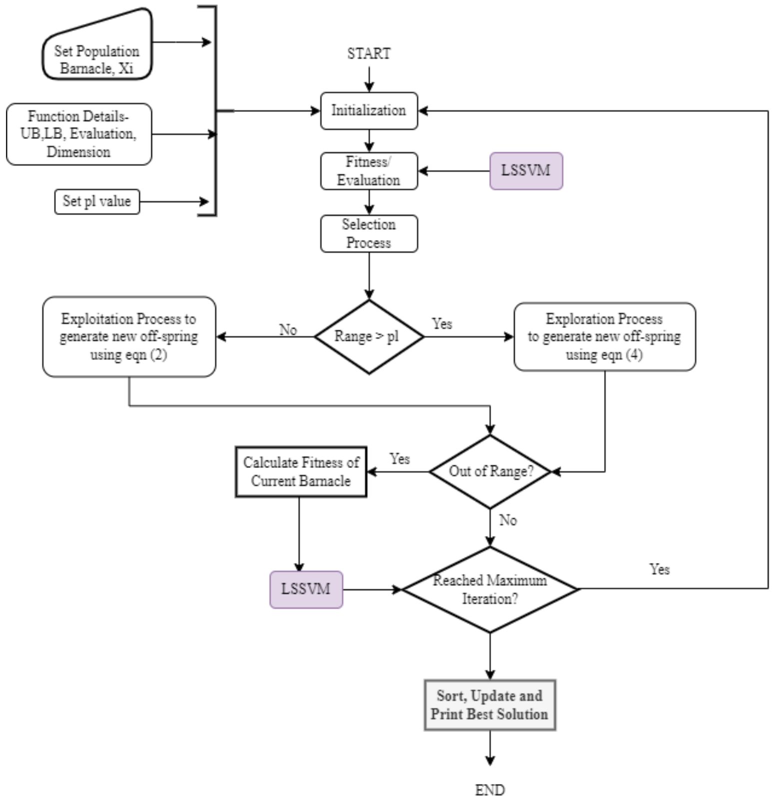


Fig. 2. Flow diagram of IBMO-LSSVM

The process flow of IBMO is started by creating an X population, as described in Equations (1)-(3) generate new offspring. It is followed by the evaluation process for each new offspring, which will then be combined with the parents. The sorting process is executed to place the current best solution at the top of the population. Only half of the top population (which consists of a mix of parents and offspring) is chosen for the evaluation of the next iteration, where the bottom half of the population is assumed to be eliminated.

The application of the IBMO Algorithm in forecasting COVID-19 situations is to determine the best combination of control variables that produces the minimum objective function evaluation. This paper's objective is to minimize prediction errors without violating any constraints. Initially, the value of pl, total population, and maximum iteration are set. It is followed by the determination of all function details, such as the data of the test system and the boundary of the searching areas. Then, each control variable for each population is stored temporarily in the data of the selected test system. After running the model using Matlab, the total predicted outcomes for each population could be obtained.

```
# Start
Initialize population barnacles (Xi)
Calculate fitness using LSSVM
Sorting the best Solution, T
While(I<Maximum iterations)
    Set pl;
    Selection process for mating
    if Dad and Mum is equal to pl
        Exploitation Process using (2)
    else if Dad and Mum is greater than pl
        Exploration Process using (4) //Improved the random walk by Levy Flight
    end if
Retrieve current barnacles when it goes out of range;
Calculate the best fitness using LSSVM;
Sort and update if T is better;
I++;
end While
Return the best solution T;
```

Fig. 3. Pseudo code of IBMO-LSSVM. Improved from [15]

4.2. Properties Setting

Formerly executing the experimental procedure, the characteristics of IBMO-LSSVM, BMO-LSSVM, BMO-GAUSS-LSSVM, SOGWO-LSSVM, PSO-LSSVM, SSA-LSSVM, and MVO-LSSVM are all configured with the same property setting as shown in Table 2. The relevant attributes include population size, maximum iterations, upper and lower bounds for LSSVM hyper parameters. All property values are determined by trial and error. The maximum number of iterations is 50. Similarly, the population size is sufficient to get the desired outcomes.

Table 2. Property setting of different algorithms

Property	Same for all algorithms
Population size	50
Maximum Iteration	50
Lower bound	1
Upper bound	1000

4.3. Performance evaluation

Since time series forecasting models have been evaluated through metrics such as, Mean Absolute Percentage Error (MAPE), Theils'U and accuracy, these performance matrices for regression are defined as follows:

(9) $$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n} |(y_{\text{predicted}} - y_{\text{actual}})/y_{\text{actual}}| \times 100\%,$$

(10) $$\text{Theils'}U = \frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(\text{y}_{\text{actual}} - \text{y}_{\text{predicted}})^2}}{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(\text{y}_{\text{actual}})^2} + \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\text{y}_{\text{predicted}})^2}},$$

(11) $$\text{Accuracy} = 1 - \text{MAPE},$$

where: n is the number of test instances; $y_{\text{predicted}}$ is the predicted values at i-th time; y_{actual} is the actual values at i-th time.

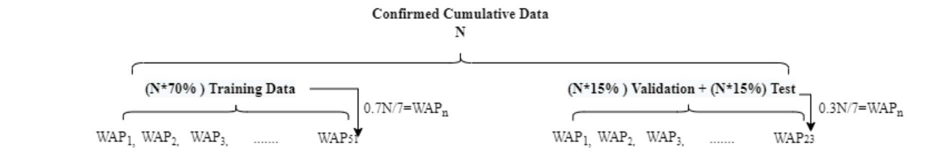
The above equations (9)-(11) are the common evaluation indicators that define the error rate of the prediction model for regression. Their values should be as small as possible.

4.4. Dataset preparation and input-output

The data is obtained from January 1, 2020, to July 27, 2022, and collected daily. It will be grouped into different sets, such as training, validation, and testing. On the website **https://data.humdata.org/dataset/novel-coronavirus-2019-ncov-casess**, you can find the entire COVID-19 validated data set. Later, the dataset is merged and updated using the link provided below for the total vaccination cases since the total vaccination count started officially on 24 February 2021, in Malaysia.

The daily collection begins on 24 February 2021, and continues until 27 July 2022. The data will then be organized into sets for training, validation, and testing. The full cumulative confirmed verified data set for COVID-19 at **https://data.humdata.org/dataset/novel-coronavirus-2019-ncov-casess**.

Fig. 4 shows the schematic diagram of the dataset. The fully vaccinated cumulative data, on the other hand, has been compiled from the website for the same period: **https://ourworldindata.org/covid-vaccinations**



Input								Output
Total Cumulative Vaccination	Cumulative Confirmed Cases							Confirmed Cases Prediction Average Point
TCV-W1D1	W1D1	W1D2	W1D3	W1D4	W1D5	W1D6	W1D7	CCP-W1AP1
TCV-W2D1	W2D1	W2D2	W2D3	W2D4	W2D5	W2D6	W2D7	CCP-W2AP2
TCV-W3D1	W3D1	W3D2	W3D3	W3D4	W3D5	W3D6	W3D7	CCP-W3AP3
TCV-W4D1	W4D1	W4D2	W4D3	W4D4	W4D5	W4D6	W4D7	CCP-W4AP4
TCV-W5D1	W5D1	W5D2	W5D3	W5D4	W5D5	W5D6	W5D7	CCP-W5AP5
TCV-W6D1	W6D1	W6D2	W6D3	W6D4	W6D5	W6D6	W6D7	CCP-W6AP6
TCV-W7D1	W7D1	W7D2	W7D3	W7D4	W7D5	W7D6	W7D7	CCP-W7AP7

Fig. 4. Schematic diagram of data modelling

The dataset is divided into three parts: training, validation, and testing, with a 70-15-15 split. All three sets-training, validation, and testing-were used to train and adjust the LSSVM hyper-parameters and models, and the final results were evaluated on all three sets.

5. Results

The experiment design that is compatible with the study has been already explained in Section 4. Moreover, the data has been divided into three phases: training, validation, and testing. Finally, there are performance evaluation criteria for the IBMO-LSSVM model.

Various common benchmark test functions are used in the literature to evaluate the performance of this method. Additionally, our results are compared to those of other well-known algorithms in the literature: PSO, GA, DA, WOA, and SSA. It is worth mentioning that the results of 19 classical benchmark test functions PSO, GA, and DA are taken from this work [33]. However, we conducted the CEC-C06 tests. Moreover, four measurement metrics have been used for further observation. Finally, the IBMO has been used for optimizing real-world applications.

5.1. Classical benchmark functions

The performance of the IBMO Algorithm is tested using three sets of test functions: unimodal, multimodal, and composite [33]. Each collection of these test functions is intended to assess certain algorithmic views. As their name would indicate, they have a single optimum; unimodal benchmark functions, for instance, are used to assess the exploitation level and convergence of the algorithm. However, since they have several optimal solutions, multimodal benchmark functions are used to assess the local optimum avoidance and exploration levels. Similar to multimodal algorithms, there are many optimum solutions; among them are the most global and local solutions. The objective of the algorithm is to avoid local optimum solutions and to converge to a globally optimal solution. In addition, most of the composite benchmark functions are merged, rotated, shifted, and biased versions of other test functions. Composite benchmark functions contain a very large number of local optima and provide a variety of forms for various search landscape areas. This kind of benchmark function shows difficulties in practical search spaces. The results of the traditional benchmark function are given in Table 3, which compares IBMO to other optimization strategies such as Fitness Dependent Optimizer (FDO), Dragonfly Algorithm (DA), Particle Swarm Optimizer (PSO), and Genetic Algorithm (GA).

Each algorithm presented in Table 3 has been put through its paces 30 times, with each test utilising 30 search agents, each of which possessed 10 dimensions, primarily; later, the dimension has been set up to 100 to improve the performances and only the best value for TF9, TF10, TF12, TF13 has been updated. During each test, the algorithm has been allowed to search for the most optimal solution throughout 500 iterations, after which the average and standard deviation have been determined. Concerning the several types of parameters sets, this work [33] describes the FDO, GA, PSO, and DA parameter sets. Every test function has been minimised towards the value of 0.0, except TF8, which has been minimised towards the value of –418. For example, several test functions have been moved away from the starting position to demonstrate that the algorithms does not have a preference for the starting point. Unimodal benchmark functions [35], Multimodal benchmark functions [35], and Composite benchmark functions [35] demonstrate the details of the explanation and parameter setting of the applied classical benchmark functions.

Table 3. Classical benchmark results of selected algorithms with IBMO [33]

Test function	IBMO		FDO		DA		PSO		GA	
	Average	STD	Average	STD	Average	STD	Average	STD	Average	STD
TF1	**2.21×10^{-35}**	**5.05×10^{-35}**	7.47×10^{-21}	7.26×10^{-19}	2.85×10^{-18}	7.16×10^{-18}	4.2×10^{-18}	1.31×10^{-17}	7.49×10^{2}	3.25×10^{2}
TF2	**2.35×10^{-8}**	**2.17×10^{-8}**	9.39×10^{-6}	6.91×10^{-6}	1.49×10^{-5}	3.76×10^{-5}	0.003154	9.81×10^{-3}	5.97×10^{0}	1.533102
TF3	1.24×10^{-1}	3.97×10^{-1}	**8.55×10^{-7}**	4.40×10^{-6}	1.29×10^{-6}	2.10×10^{-6}	1.89×10^{-3}	3.31×10^{-3}	1.95×10^{3}	9.94×10^{2}
TF4	4.20×10^{0}	5.92×10^{0}	**6.69×10^{-4}**	2.49×10^{-3}	9.88×10^{-4}	2.78×10^{-3}	1.75×10^{-3}	0.002515	2.12×10^{1}	2.61×10^{0}
TF5	7.55×10^{1}	1.20×10^{2}	2.35×10^{1}	5.98×10^{1}	7.60×10^{0}	6.79×10^{0}	6.35×10^{1}	**80.12726**	1.33×10^{5}	8.50×10^{4}
TF6	1.84×10^{-12}	2.36×10^{-12}	**1.42×10^{-18}**	4.75×10^{-18}	4.17×10^{-16}	1.32×10^{-15}	4.36×10^{-17}	1.38×10^{-16}	5.64×10^{2}	2.30×10^{2}
TF7	**8.76×10^{-3}**	**3.29×10^{-3}**	5.44×10^{-1}	3.15×10^{-1}	1.03×10^{-2}	4.69×10^{-3}	5.97×10^{-3}	3.58×10^{-3}	1.67×10^{-1}	0.072571
TF8	**-2.87×10^{3}**	**3.90×10^{2}**	-2.29×10^{6}	2.07×10^{5}	-2.86×10^{3}	3.84×10^{2}	-7.1×10^{11}	1.20×10^{12}	-3.41×10^{3}	1.64×10^{2}
TF9	3.04×10^{1}	1.54×10^{1}	1.46×10^{1}	**5.20×10^{0}**	1.60×10^{1}	9.48×10^{0}	1.04×10^{1}	7.88×10^{0}	25.51886	6.67×10^{0}
TF10	3.50×10^{0}	6.41×10^{0}	4.00×10^{-15}	6.38×10^{-16}	2.31×10^{-1}	4.87×10^{-1}	2.80×10^{-1}	6.02×10^{-1}	9.50×10^{0}	1.27×10^{0}
TF11	1.82×10^{-1}	1.01×10^{-1}	5.69×10^{-1}	1.04×10^{-1}	1.93×10^{-1}	7.35×10^{-2}	**8.35×10^{-2}**	**3.51×10^{-2}**	7.72×10^{0}	3.63×10^{0}
TF12	6.43×10^{-1}	1.24×10^{0}	1.98×10^{1}	2.64×10^{1}	3.11×10^{-2}	9.83×10^{-2}	**8.57×10^{-11}**	**2.71×10^{-10}**	1.86×10^{3}	5.82×10^{3}
TF13	3.66×10^{-3}	5.27×10^{-3}	1.03×10^{1}	7.42×10^{0}	2.20×10^{-3}	4.63×10^{-3}	**2.20×10^{-3}**	**4.63×10^{-3}**	6.80×10^{4}	8.77×10^{4}
TF14	4.57×10^{0}	2.99×10^{0}	**3.79×10^{-7}**	**6.32×10^{-7}**	1.04×10^{2}	9.12×10^{1}	1.50×10^{2}	1.35×10^{2}	130.0991	2.13×10^{1}
TF15	**2.18×10^{-3}**	4.95×10^{-3}	**1.50×10^{-3}**	1.24×10^{-3}	1.93×10^{2}	8.06×10^{1}	1.88×10^{2}	1.57×10^{2}	1.16×10^{2}	1.92×10^{1}
TF16	-1.03×10^{0}	0.00×10^{0}	**6.38×10^{-3}**	**1.06×10^{-2}**	4.58×10^{2}	1.65×10^{2}	2.63×10^{2}	1.87×10^{2}	3.84×10^{2}	3.66×10^{1}
TF17	**3.98×10^{-1}**	**1.13×10^{-16}**	2.38×10^{1}	2.15×10^{-1}	5.97×10^{2}	1.71×10^{2}	4.67×10^{2}	1.81×10^{2}	5.03×10^{2}	3.58×10^{1}
TF18	**3.00×10^{0}**	**4.81×10^{-15}**	2.23×10^{2}	9.96×10^{-6}	2.30×10^{2}	1.85×10^{2}	1.36×10^{2}	1.60×10^{2}	1.18×10^{2}	51.00.183
TF19	**-3.86×10^{0}**	**2.71×10^{-15}**	2.28×10^{1}	1.04×10^{-2}	6.80×10^{2}	1.99×10^{2}	741.6341	2.07×10^{2}	5.44×10^{2}	13.30161

The outcomes of IBMO, FDO, DA, PSO, and GA, are detailed in Table 3. In general, the findings reveal that IBMO produces superior results to TF1, TF2, TF7, TF8, TF15, TF17, TF18, and TF19 algorithms, although the results for other test functions show that the other algorithms wereare superior. Despite having better outcomes than GA, IBMO has poor performances in TF5 and TF6 compared to others. In the cases of TF5, TF9, TF10, TF12, TF13, and TF14, IBMO, on the other hand, delivered results are substantially comparable to those produced by the other algorithms. However, the findings of TF1, TF2, TF7, TF8, TF15, TF17, TF18, and TF19 demonstrate that the IBMO algorithm performes better than FDA, DA, PSO, and GA in every scenario.

5.2. CEC-C06 2019 benchmark test functions

A collection of 10 current CEC benchmark test functions is utilized to evaluate IBMO. These test functions have been enhanced by Professor Suganthan and his team [34]. The 100-Digit Challenge tests are utilized in an annual optimization competition. The CEC benchmark developer specified CEC04 to CEC10 as a 10-dimensional minimization issue in the [–100, 100] border range. However, CEC01 to CEC03 has different dimensions, as stated in [33]. The test functions CEC04 to CEC10 are shifted and rotated; CEC01 to CEC03 are not. All tests may be scaled. CEC-C06 2019 Benchmarks "The 100-Digit Challenge": [33] describes the published function details designed by the CEC functions competition developer. However, 1000 dimension has been applied to few functions to check the results and only CEC01 results has been updated accordingly.

In this paper, IBMO competes with the following algorithms: FDO, DA, WOA, and SSA in terms of the defined CEC functions. These algorithms are highly-cited in the literature and perform well on benchmark tests and real-world applications. Authors publish their algorithm implementations. Regarding algorithm parameter settings, their defaults have not been changed throughout the testing; all competitors have used the values from their original publications [35, 38].

Each algorithm has 500 iterations and 30 agents. IBMO beats other algorithms in CEC03, CEC05, CEC06, and CEC08. FDO and SSA have superior outcomes for CEC01, CEC04, and CEC07, while only FDO is similar for CEC02, CEC09, and CEC10. CEC09 has similar results for IBMO and FDO. Table 4 displays the results of 10 CEC functions and their comparison to the chosen algorithms and IBMO, where Levy-Flights enhance BMO.

Table 4. IEEE ECE 2019 benchmark results

Test function	IBMO		FDO		DA		WOA		SSA	
	Average	STD	Average	STD	Average	STD	Average	STD	Average	STD
CEC01	1.48×10^{10}	1.3×10^{10}	4585.27	20707.63	543×10^{8}	669×10^{8}	411×10^{8}	542×10^{8}	605×10^{7}	475×10^{7}
CEC02	17.34286	0	4	3.22×10^{-9}	78.0368	87.7888	17.3495	0.0045	18.3434	0.0005
CEC03	12.7024	1.25×10^{-8}	13.7024	1.65×10^{-11}	13.7026	0.0007	13.7024	0	13.7025	0.0003
CEC04	95.99308	106.1765	34.0837	16.52887	344.3561	414.0982	394.6754	248.5627	41.6936	22.2191
CEC05	1.285901	0.194502	2.13924	0.085751	2.5572	0.3245	2.7342	0.2917	2.2084	0.1064
CEC06	4.417496	2.155916	12.1332	0.600237	9.8955	1.6404	10.7085	1.0325	6.0798	1.4873
CEC07	447.1324	270.25	120.4858	13.59369	578.9531	329.3983	490.6843	1948318	410396	290.556
CEC08	5.791751	0.413281	6.1021	0.756997	6.8734	0.5015	6.909	0.4269	6.37	0.5862
CEC09	2.743075	0.329955	2	1.59×10^{-10}	6.0467	2.871	5.9371	1.6566	3.6704	0.2362
CEC010	20.10046	0.141054	2.7182	8.88×10^{-16}	21.2604	0.1715	21.2761	0.1111	21.04	0.078

We have summarized the performance comparison of the CEC function from Table 4 to comprehend its position in Fig. 5 for clear understanding. In the first row, IBMO surpasses other mentioned algorithms; however, in the second and third row, IBMO's performance is comparable to other algorithms.

CEC03	
IBMO	**12.702404**
FDO	13.7024
WOA	13.7024
SSA	13.7025
DA	13.7026

CEC05	
IBMO	**1.2859**
FDO	2.13924
SSA	2.2084
DA	2.5572
WOA	2.7342

CEC06	
IBMO	**4.4174955**
SSA	6.0798
DA	9.8955
WOA	10.7085
FDO	12.1332

CEC08	
IBMO	**5.791751**
FDO	6.1021
SSA	6.37
DA	6.8734
WOA	6.909

CEC02	
FDO	4
IBMO	**17.34285714**
WOA	17.3495
SSA	18.3434
DA	78.0368

CEC09	
FDO	2
IBMO	**2.7430752**
SSA	3.6704
WOA	5.9371
DA	6.0467

CEC10	
FDO	2.7182
IBMO	**20.1004595**
SSA	21.04
DA	21.2604
WOA	21.2761

CEC01	
FDO	4585.27
SSA	6.05E+09
IBMO	**1.48E+10**
WOA	4.11E+10
DA	5.43E+10

CEC04	
FDO	34.0837
SSA	41.6936
IBMO	**95.993077**
DA	344.3561
WOA	394.6754

CEC07	
FDO	120.4858
SSA	410.3964
IBMO	**447.132413**
WOA	490.6843
DA	578.9531

Fig. 5. CEC functions results comparison

5.3. IBMO real world application

IBMO may be used to address application-specific challenges in real-world settings, much like any other metaheuristic algorithm. IBMO is used to forecast the time series of confirmed COVID-19 cases in this part in conjunction with LSSVM. This paper summarizes the acquired findings when performing the series of analyses. Table 5 shows the measured and estimated outcomes using all the hybrid algorithms implemented in these studies compared with the target values of confirmed cases considering the total vaccination.

Table 5. Target vs IBMO-LSSVM, BMO-LSSVM, SSA-LSSVM, MVO-LSSVM, PSO-LSVM, SOGWO-LSSVM and ANN

Week	Target	**IBMO-LSSVM**	BMO-LSSVM	BMO-Gauss-LSSVM	ANN	SOGVO-LSSVM	PSO-LSSVM	MVO-LSSVM	SSA-LSSVM
67	4502155	**4495851.983**	4448579.356	4486397.458	400691795	4486397.458	4480994.872	4386449.617	4401306.728
68	4513277.286	**4506958.698**	4459569.286	4497480.815	4016816.785	4497480.815	4492064.883	4397286.06	4412179.875
69	4524651.714	**4518317.202**	4470808.359	4508815.433	4026940.025	4508815.433	4503385.851	4408368.165	4423299.516
70	4538708.143	**4532353.952**	4484697.516	4522822.664	4039450.247	4522822.664	4517376.215	4422063.344	4437041.081
71	4554246.143	**4547870.198**	4500050.614	4538306.281	4053279.067	4538306.281	4532841.186	4437202.017	4452231.029
72	4571123.857	**4564724.284**	4516727.483	4555124.924	4068300.233	4555124.924	4549639.575	4453645.974	4468730.683
73	4592335.143	**4585905.874**	4537686.355	4576261.97	4087178.277	4578261.97	4570751.168	4474312.13	4489466.836
74	4617926.571	**4611461.474**	4562973.245	4601763.828	4109954.648	4601763.828	4596222.316	4499245.858	4514485.016
75	4647891.571	**4641384.523**	4592581.661	4631623.951	4136623.498	4631623.951	4626046.481	4528440.758	4543778.8

The prediction accuracy is 99.86%. Table 6 indicates that IBMO-LSSVM surpasses the competition by delivering the lowest MAPE value, 0.0013754. The performance of the other metrics, Theils'U and accuracy, obtained by IBMO-LSSVM is also superior to that of the other hybrid models, including the original and one variant of BMO. The table highlights that IBMO-LSSVM outperforms by producing the lowest value for all performance matrices.

Table 6. Performance evaluation of different algorithms

Performance comparison			
Comparison among BMO variants			
Algorithm	MAPE	Accuracy	Theils'U
IBMO-LSSVM	0.0013754	0.9986246	0.0014
BMO-LSSVM	0.011887	0.988113	0.011899999
BMO-Gauss-LSSVM	0.00354	0.99646	0.006987749
Comparison with other algorithms	MAPE	Accuracy	Theils'U
SSA-LSSVM	0.0224	0.9776	0.022399998
SOGWO-LSSVM	0.0035	0.9965	0.0035
MVO-LSSVM	0.02574	0.97426	0.025699997
PSO-LSSVM	0.004655	0.995345	0.004699999
ANN	0.11	0.89	0.109999989

Fig. 6 depicts the above statistical summarized outcomes graphically. It is clear from here, that the proposed model produces more accurate prediction results daily through considering the cumulative confirmed cases and total vaccination in testing phase, compared to other hybrid algorithms.

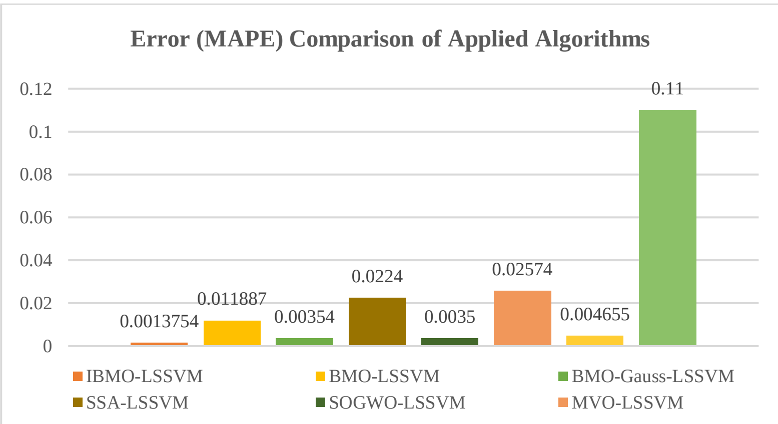


Fig. 6. MAPE comparison among algorithms

Fig. 7 shows the comparison between the target and the predicted values. The confirmed case prediction with total vaccination in Malaysia through IBMO-LSSVM, BMO-Gauss-LSSVM, and BMO-LSSVM, where IBMO-LSSVM is superior to the original BMO-LSSVM and variants such as BMO-Gauss-LSSVM in terms of performance.

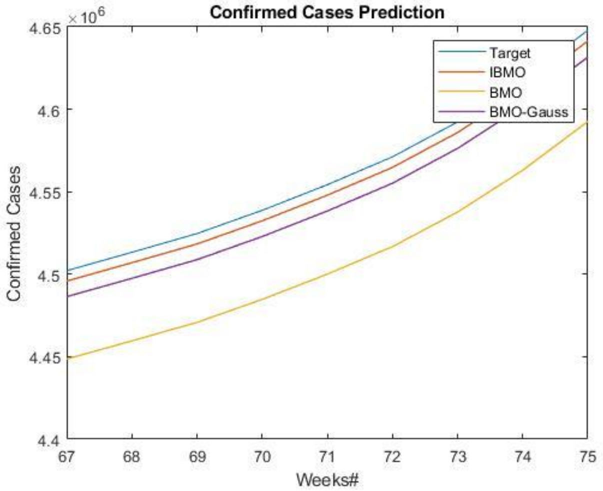


Fig. 7. Confirmed cases prediction: Target vs IBMO-LSSVM, and vs BMO-LSSVM, and vs BMO-Gauss-LSSVM

Similar to Fig. 7, Fig. 8 shows a comparison of the performance of IBMO-LSSVM in comparison to other chosen algorithms. Again, IBMO-LSSVM performes exceptionally well in comparison to the other algorithms.

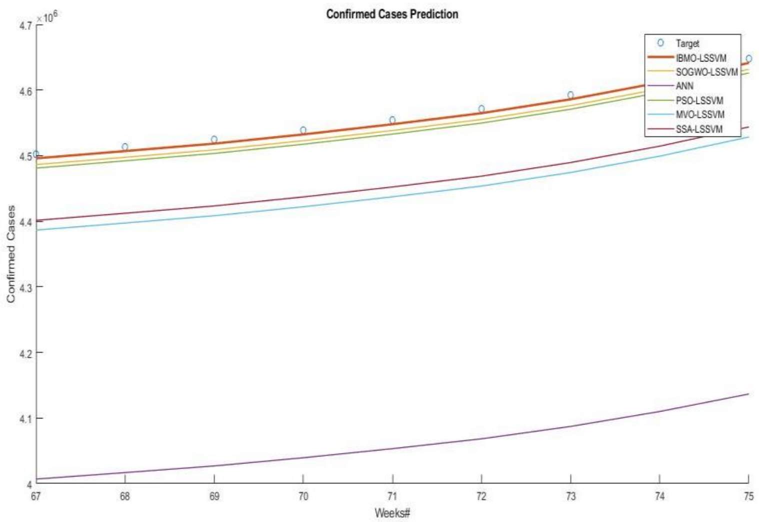


Fig. 8. Confirmed cases prediction: Target vs IBMO-LSSVM, and vs SOGWO-LSSVM, and vs PSO-LSSVM, and vs MVO-LSSVM, and vs SSA-LSSVM, and vs ANN

6. Conclusion

The improvement of recent bio-inspired algorithms, IBMO, has been proposed as a novel solution for forecasting the weekly average of confirmed cases for COVID-19 or any other infectious disease with consideration to total vaccination rates. Improvements have been made to enhance the exploration process of BMO, where the implementation of Levy flight has been enforced and substituted in the sperm cast equation of the original BMO. The performance and effectiveness of the improved algorithm have been tested with other algorithms. Furthermore, the IBMO has been evaluated using a collection of 19 different single-objective benchmark testing functions. The functions of benchmark testing have been partitioned into three separate categories: unimodal, multimodal, and composite. In addition, IBMO has conducted tests on ten current CEC-C06 benchmarks. When the results of IBMO are compared to those of two well-known algorithms (PSO and GA) and four modern algorithms (FDO, DA, WOA, and SSA), the findings has shown that IBMO beats the competing algorithms in the majority of the situations and provides comparable results in the rest. For statistical significance, the Theil'sU test has been used. In all, four individual tests has been carried out on the IBMO algorithm to assess, demonstrate, and verify its performance and credibility. Moreover, the IBMO method is shown to be capable of addressing real-life problems by applying it in practice to one case taken from the real world, the primary concern of WHO. The IBMO is used in this work to adjust the LSSVM hyper-parameters. Later, considering the total number of vaccination cases, LSSVM will forecast COVID-19 confirmed cases using the optimal values predicted by the IBMO. IBMO-LSSVM has been compared against various state-of-the-art hybrid algorithms using the same property setup, including SOGWO-LSSVM, MVO-LSSVM, PSO-LSSVM, SSA-LSSVM, ANN's original BMO-LSSVM, and variations of BMO (BMO-Gauss-LSSVM). When compared to its competitors, IBMO-LSSVM consistently achieves better results.

***Acknowledgments*:** This research study was supported by Ministry Of Education Malaysia (MOE) and University Malaysia Pahang under Fundamental Research Grant Scheme (FRGS/1/2019/ICT02/UMP/03/1) & (#RDU1901133).

References

1. L a n d i, F., et al. The New Challenge of Geriatrics: Saving Frail Older People from the SARS-COV-2 Pandemic Infection. – Journal of Nutrition, Health and Aging, Vol. **24**, 2020, No 5, pp. 466-470. DOI: 10.1007/s12603-020-1356-x.
2. L a n d i, F., et al. Post-COVID-19 Global Health Strategies: The Need for an Interdisciplinary Approach. – Aging Clinical and Experimental Research, Vol. **32**, 2020, No 8, pp. 1613-1620. DOI: 10.1007/s40520-020-01616-x.
3. C h e n g, Z. J., J. S h a n. 2019 Novel Coronavirus: Where We Are and What We Know. – Infection, Vol. **48**, 2020, No 2, pp. 155-163. DOI: 10.1007/s15010-020-01401-y.
4. G u a n, W. J., et al. China Medical Treatment Expert Group for COVID-19. Clinical Characteristics of Coronavirus Disease 2019 in China. – N. Engl. J. Med., Vol. **382**, 30 April 2020, No 18, pp. 1708-1720. DOI: 10.1056/NEJMoa2002032. Epub 2020 Feb 28. PMID: 32109013, PMCID: PMC7092819.

5. Quintana, V. H., M. Santos-Nieto. Reactive-Power Dispatch by Successive Quadratic Programming. – IEEE Transactions on Energy Conversion, Vol. **4**, 1989, No 3, pp. 425-435. DOI: 10.1109/60.43245.
6. Aceves-Lara, C. A., et al. Kinetic Parameters Estimation in an Anaerobic Digestion Process Using Successive Quadratic Programming. – Water Science and Technology, Vol. **52**, 2005, No 1-2, pp. 419-426. DOI: 10.2166/wst.2005.0548.
7. Xie, Y. F. Reduced Hessian Successive Quadratic Programming Algorithm. – Large-Scale Optimization with Applications, Part I: Optimization in Inverse Problems and Design, Vol. **92**, p. 195.
8. Wan, C., J. Wang, J. Lin, Y. Song, Z. Y. Dong. Nonparametric Prediction Intervals of Wind Power via Linear Programming. – IEEE Transactions on Power Systems, Vol. **33**, 2018, No 1, pp. 1074-1076. DOI: 10.1109/TPWRS.2017.2716658.
9. Lee, E. K., T. L. Wu. Classification and Disease Prediction via Mathematical Programming. – Data Mining, Systems Analysis and Optimization in Biomedicine, Vol. **953**. 2007. DOI: 10.1063/1.2817343.
10. Cuéllar, M. P., M. Delgado, M. C. Pegalajar. An Application of Non-Linear Programming to Train Recurrent Neural Networks in Time Series Prediction Problems. – In: Proc. of 7th International Conference on Enterprise Information Systems (ICEIS'2005), 2005, No Lm, pp. 35-42. DOI: 10.1007/978-1-4020-5347-4_11.
11. Mangasarian, O. L., W. N. Street, W. H. Wolberg. Breast Cancer Diagnosis and Prognosis via Linear Programming. – Operations Research, Vol. **43**, 1995, No 4, pp. 570-577. DOI: 10.1287/opre.43.4.570.
12. Zarrin, P., M. Maleki, Z. Khodadai, R. B. Arellano-Valle. Time Series Models Based on the Unrestricted Skew-Normal Process. – Journal of Statistical Computation and Simulation, Vol. **89**, 2019, No 1, pp. 38-51. DOI: 10.1080/00949655.2018.1533962.
13. Ghasami, S., M. Maleki, Z. Khodadadi. Leptokurtic and Platykurtic Class of Robust Symmetrical and Asymmetrical Time Series Models. – Journal of Computational and Applied Mathematics, Vol. **376**, 2020, p. 112806. DOI: 10.1016/j.cam.2020.112806.
14. Maleki, M., A. R. Nematollahi. Autoregressive Models with Mixture of Scale Mixtures of Gaussian Innovations. – Iranian Journal of Science and Technology, Transaction A: Science, Vol. **41**, 2017, No 4, pp. 1099-1107. DOI: 10.1007/s40995-017-0237-6.
15. Mustaffa, Z., M. H. Sulaiman. COVID-19 Confirmed Cases Prediction in China Based on Barnacles Mating Optimizer-Least Squares Support Vector Machines. – Cybernetics and Information Technologies, Vol. **21**, 2021, No 4, pp. 62-76.
16. Mustaffa, Z., H. Sulaiman, K. A. M. Rosli, M. F. M. Mohsin, Y. Yusof. Predictive Analysis of Dengue Outbreak Based on an Improved Salp Swarm Algorithm. – Cybernetics and Information Technologies, Vol. **20**, 2020, No 4, pp. 156-169.
17. Li, G., K. Chen, H. Yang. A New Hybrid Prediction Model of Cumulative COVID-19 Confirmed Data. – Process Safety and Environmental Protection, Vol. **157**, 2022, pp. 1-19. DOI: 10.1016/j.psep.2021.10.047.
18. Maleki, M., M. R. Mahmoudi, D. Wraith, K. H. Pho. Time Series Modelling to Forecast the Confirmed and Recovered Cases of COVID-19. – Travel Medicine and Infectious Disease, Vol. **37**, 2020, No March, p. 101742. DOI: 10.1016/j.tmaid.2020.101742.
19. Talkhi, N., N. Akhavan Fatemi, Z. Ataei, M. Jabbari Nooghabi. Modeling and Forecasting Number of Confirmed and Death Caused COVID-19 in IRAN: A Comparison of Time Series Forecasting Methods. – Biomedical Signal Processing and Control, Vol. **66**, 2021, No November 2020, p. 102494. DOI: 10.1016/j.bspc.2021.102494.
20. Chimmula, V. K. R., L. Zhang. Time Series Forecasting of COVID-19 Transmission in Canada Using LSTM Networks. – Chaos, Solitons and Fractals, Vol. **135**, 2020. DOI: 10.1016/j.chaos.2020.109864.
21. Zeroual, A., F. Harrou, A. Dairi, Y. Sun. Deep Learning Methods for Forecasting COVID-19 Time-Series Data: A Comparative Study. – Chaos, Solitons and Fractals, Vol. **140**, 2020, p. 110121. DOI: 10.1016/j.chaos.2020.110121.
22. Shastri, S., K. Singh, S. Kumar, P. Kour, V. Mansotra. Time Series Forecasting of COVID-19 Using Deep Learning Models: India-USA Comparative Case study. – Chaos, Solitons and Fractals, Vol. **140**, 2020, p. 110227. DOI: 10.1016/j.chaos.2020.110227.

23. K u m a r, N., S. S u s a n. COVID-19 Pandemic Prediction Using Time Series Forecasting Models. – In: Proc. of 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT'20), 2020. DOI: 10.1109/ICCCNT49239.2020.9225319.
24. A l a s s a f i, M. O., M. J a r r a h, R. A l o t a i b i. Time Series Predicting of COVID-19 Based on Deep Learning. – Neurocomputing, Vol. **468**, 2022, pp. 335-344. DOI: 10.1016/j.neucom.2021.10.035.
25. C l e r c, M. Particle Swarm Optimization. – Particle Swarm Optimization, 2010, pp. 1942-1948. DOI: 10.1002/9780470612163.
26. G o l d, J. E., R. A. O k y a y, W. E. L i c h t, D. J. H u r l e y. Investigation of Long Covid Prevalence and Its Relationship to Epstein-Barr Virus Reactivation. – Pathogens, Vol. **10**, 2021, No 6, pp. 1-15. DOI: 10.3390/pathogens10060763.
27. M i r j a l i l i, S., S. M. M i r j a l i l i, A. L e w i s. Grey Wolf Optimizer. – Advances in Engineering Software, Vol. **69**, 2014, pp. 46-61. DOI: 10.1016/j.advengsoft.2013.12.007.
28. M i r j a l i l i, S. Moth-Flame Optimization Algorithm: A Novel Nature-Inspired Heuristic Paradigm. – Knowledge-Based Systems, Vol. **89**, 2015, pp. 228-249. DOI: 10.1016/j.knosys.2015.07.006.
29. COVID Live – Coronavirus Statistics – Worldometer. (Accessed 14 June 2022). **https://www.worldometers.info/coronavirus/**
30. S u l a i m a n, M. H., Z. M u s t a f f a, M. M. S a a r i, H. D a n i y a l. Barnacles Mating Optimizer: A New Bio-Inspired Algorithm for Solving Engineering Optimization Problems. – Engineering Applications of Artificial Intelligence, Vol. **87**, 2020, pp. 265-270. DOI: 10.1016/j.engappai.2019.103330.
31. S u l a i m a n, M. H., Z. M u s t a f f a, M. M. S a a r i, H. D a n i y a l, I. M u s i r i n, M. R. D a u d. Barnacles Mating Optimizer: An Evolutionary Algorithm for Solving Optimization. – In: Proc. of IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS'18), October 2018, pp. 99-104. DOI: 10.1109/I2CACIS.2018.8603703.
32. S u l a i m a n, M. H., et al. Barnacles Mating Optimizer: A Bio-Inspired Algorithm for Solving Optimization Problems. – In: Proc. of 19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD'18), Vol. **87**, June 2018, No September 2019, pp. 265-270. DOI: 10.1109/SNPD.2018.8441097.
33. A b d u l l a h, J. M., T. A h m e d. Fitness Dependent Optimizer: Inspired by the Bee Swarming Reproductive Process. – IEEE Access, Vol. **7**, 2019, pp. 43473-43486.
34. P r i c e, K. V., N. H. A w a d, M. Z. A l i, P. N. S u g a n t h a n. The 100-Digit Challenge: Problem de_nitions and Evaluation Criteria for the 100-Digit Challenge Special Session and Competition on Single Objective Numerical Optimization. – School Elect. Electron. Eng., Nanyang Technol. Univ. of Singapore, Tech. Rep., November 2018.
35. M i r j a l i l i, S. Dragon_y Algorithm: A New Meta-Heuristic Optimization Technique for Solving Single-Objective, Discrete, and Multi-Objective Problems. – Neural Comput. Appl., Vol. **27**, May 2015, No 4, pp. 1053-1073.
36. M i r j a l i l i a b, S., A. L e w i s a. The Whale Optimization Algorithm. – Adv. Eng. Softw., Vol. **95**, May 2016, pp. 51-67.
37. M i r j a l i l i a, S., A. H. G a n d o m i b f, S. Z. M i r j a l i l i, C. S a r e m i a, H. F a r i s d, S. M. M i r j a l i l i e. Salp Swarm Algorithm: A Bio-Inspired Optimizer for Engineering Design Problems. – Adv. Eng. Softw., Vol. **114**, December 2017, pp. 163-191.
38. M i r j a l i l i, A., S. M i r j a l i l i. Seyedali Mirjalili. 2015 (Accessed 01 January 2019). **http://www.alimirjalili.com/Projects.html**

Received: 04.10.2022; Second Version: 31.01.2023; Accepted: 10.02.2023