



Contents lists available at ScienceDirect

## Journal of King Saud University - Science

journal homepage: [www.sciencedirect.com](http://www.sciencedirect.com)

Full Length Article



# Cloud spot instance price forecasting multi-headed models tuned using modified PSO

Mohamed Salb<sup>a</sup>, Luka Jovanovic<sup>a</sup>, Ali Elsadai<sup>a</sup>, Nebojsa Bacanin<sup>a,b,c</sup>, Vladimir Simic<sup>d,e,f</sup>, Dragan Pamucar<sup>g,\*</sup>, Miodrag Zivkovic<sup>a</sup>

<sup>a</sup> Singidunum University, Danijelova 32, Belgrade, 11000, Serbia

<sup>b</sup> Department of Mathematics, Saveetha School of Engineering, SIMATS, Thandalam, Chennai, Tamilnadu, 602105, India

<sup>c</sup> MEU Research Unit, Middle East University, Amman, Jordan

<sup>d</sup> University of Belgrade, Faculty of Transport and Traffic Engineering, Vojvode Stepe 305, Belgrade, 11010, Serbia

<sup>e</sup> Yuan Ze University, College of Engineering, Department of Industrial Engineering and Management, Taoyuan City, 320315, Taiwan

<sup>f</sup> Department of Computer Science and Engineering, College of Informatics, Korea University, Seoul, 02841, Republic of Korea

<sup>g</sup> Széchenyi István University, Győr, Hungary

## ARTICLE INFO

Dataset link: <https://www.kaggle.com/datasets/berkayalan/aws-ec2-instance-comparison>

## Keywords:

Cloud computing  
Gated recurrent units  
Long short term memory  
Particle swarm optimization  
Spot price forecasting

## ABSTRACT

The increasing dependence and demands on cloud infrastructure have brought to light challenges associated with cloud instance pricing. The often unpredictable nature of demand as well as changing costs of supplying a reliable instance can leave companies struggling to appropriately budget to support a healthy cash flow while maintaining operating costs. This work explores the potential of multi-headed recurrent architectures to forecast cloud instance prices based on historical and instance data. Two architectures are explored, long short-term memory (LSTM) and gated recurrent unit (GRU) networks. A modified optimizer is introduced and tested on a publicly available Amazon elastic compute cloud dataset. The GRU model, enhanced by the proposed modified approach, had the most impressive outcomes with an MAE score of 0.000801. Results have undergone meticulous statistical validation with the best-performing models further analyzed using explainable artificial intelligence techniques to provide further insight into model reasoning and information on feature importance.

## 1. Introduction

Cloud computing has risen in demand for hosting applications that deploy a large number of processing resources. Customers may face significant leasing rates as a result of this. Precise pricing forecasting is critical for consumers of cloud services (Dadashov et al., 2014). By predicting the demand, Artificial intelligence (AI) and machine learning (ML) can aid in scaling resources up or down automatically to deal with different loads properly (Yazdi and Komasi, 2024; Younas et al., 2024). ML models can also be used for cost prediction and management, allowing better budget management.

A challenge all ML models face is the necessity to tune their hyperparameters, which is widely regarded as an NP-hard challenge. Metaheuristics algorithms have proven to be capable of reaching satisfactory solutions. The research proposed in this article aims to improve cloud spot price prediction accuracy by tuned multi-headed gated recurrent units (GRU) (Dey and Salem, 2017) and long short-term memory (LSTM) (Greff et al., 2016) deep learning models, as part of

a developed ML framework. These architects are chosen due to their unique ability to address time-series forecasting well due to intrinsic memory retention mechanisms. A modified variant of particle swarm optimization (PSO) (Kennedy and Eberhart, 1995) is also introduced. The objectives of this study are:

- A modified PSO is proposed to address observed drawbacks.
- Integrated into ML framework for cloud spot instance price forecasting.
- The performance is assessed using real-world datasets and compared to other sophisticated metaheuristics.
- Shapley additive explanations (SHAP) (Lundberg and Lee, 2017) is used to analyze top model findings and evaluate feature usage.

The reminder of this work is organized as: Section 2 reviews relevant literature and cloud computing architecture. Section 3 outlines the proposed method, including modifications. Section 4 displays the

\* Corresponding author at: Széchenyi István University, Győr, Hungary.

E-mail addresses: [mohamed.salb.16@singimail.rs](mailto:mohamed.salb.16@singimail.rs) (M. Salb), [ljovanovic@singidunum.ac.rs](mailto:ljovanovic@singidunum.ac.rs) (L. Jovanovic), [aelsadai@singidunum.ac.rs](mailto:aelsadai@singidunum.ac.rs) (A. Elsadai), [nbacanin@singidunum.ac.rs](mailto:nbacanin@singidunum.ac.rs) (N. Bacanin), [vsima@sf.bg.ac.rs](mailto:vsima@sf.bg.ac.rs) (V. Simic), [dpamucar@gmail.com](mailto:dpamucar@gmail.com) (D. Pamucar), [mzivkovic@singidunum.ac.rs](mailto:mzivkovic@singidunum.ac.rs) (M. Zivkovic).

<https://doi.org/10.1016/j.jksus.2024.103473>

Received 27 June 2024; Received in revised form 27 September 2024; Accepted 28 September 2024

Available online 9 October 2024

1018-3647/© 2024 The Authors. Published by Elsevier B.V. on behalf of King Saud University. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

dataset, experiments, and outcomes. Finally, Section 5 conclusion evaluates facts and research suggestions.

## 2. Background and literature review

Cloud computing gives consumers easy access to a combined reservoir of computer power managed by service providers (Mell et al., 2011). Spot pricing in auction-based public clouds is decided on consumers' needs, making forecasting spot prices challenging (Singh and Dutta, 2015). Previous research of Ben-Yehuda and his colleagues (Agmon Ben-Yehuda et al., 2013), sought to reveal Amazon's pricing methods for underutilized EC2 capacity. The current scheduling methods include a variety of statistical techniques to help Cloud consumers identify shifts in spot pricing (Heidari et al., 2023; Heidari and Jafari Navimipour, 2022; Norozpour and Darbandi, 2020). Novel research in the field helps handle cloud computing in more efficient ways (Darbandi et al., 2018, 2012; Darbandi, 2017a). Cloud services have gained much from recent works as well (Darbandi, 2017b; Heidari et al., 2024a)

As per the "no free lunch"(NFL) (Wolpert and Macready, 1997), a single approach is equally suitable to all challenges. Additionally, metaheuristics algorithms have been effectively utilized in a variety of NP-hard challenges, including brain tumor designation from MRI scans (Jovanovic et al., 2022b), COVID-19 infection forecasting (Bacanin et al., 2023; Zivkovic et al., 2022), neural network and hyperparameters' optimization (Jovanovic et al., 2022a; Bacanin et al., 2022), recognizing credit card fraud (Jovanovic et al., 2022). Further applications include (Heidari et al., 2024b)

### 2.1. Multi-headed recurrent architecture

A multi-headed artificial neural network (Bu and Cho, 2020) is a uniquely designed network architecture with several branches or "heads". Multi-headed networks are best for processing multiple data sources at once. Sequential data is commonly handled by LSTMs. It solves fading gradients with memory cells and gate controllers. Three gates (input, forget, and output) formed the LSTM design. These gates control unit information flow. The LSTM design helps it preserve long-term dependencies better than RNNs.

A GRU network is a recurrent architecture variation with fewer parameters and computational complexity than LSTM networks. Yet, it performs similarly in many applications. Only reset and update gates were used in the GRU architecture. This approach lets the GRU capture sequential data dependencies without the expense of LSTMs. In multi-headed designs, LSTM and GRU cells allow network heads to handle complex temporal relationships in data discreetly, reducing computational expenses and making the network more resistant to overfitting.

## 3. Proposed method

### 3.1. Modified PSO algorithm

The initialization equation for PSO typically generates random agent positions and velocities. Equation representation:

$$x_{i,d}(0) = rnd() \times (max - min) + min \quad (1)$$

The symbol  $x$  represents the starting coordinates of particle  $i$  in dimension  $d$ , denoted as  $x_{i,d}(0)$ . The function  $rnd()$  produces a random decimal number inside the bounds of 0 to 1. The terms  $max$  and  $min$  provide the upper and lower limits for the search area in dimension  $d$ .

Particles measure speed and move in every iteration. The particle's ideal prior location is indicated by  $p_{i,d}$ , while the particle's global optimal location is denoted by  $p_{g,d}$ . The present speed of the  $i$ th particle's  $d$ -th dimension at time  $t$  may be determined using this equation:

$$v_{i,d}(t) = w \times v_{i,d}(t-1) + c_1 \times rnd_1 \times (p_{i,d}(t-1) - x_{i,d}(t-1)) + c_2 \times rnd_2 \times (p_{g,d} - x_{i,d}(t-1)) \quad (2)$$

$$v_{i,d} \in [-v_{max}, v_{max}] \quad (3)$$

where this equation has several parts. The  $rnd$  function demonstrates an arbitrary number produced within the range of  $[0, 1]$ . Individual and social learning features are represented by the variables  $c_1$  and  $c_2$ . The inertia weight is represented by  $w$ , which is employed to balance local and global intensification. The particle's speed is restricted to a range of  $[-v_{max}, v_{max}]$ , where  $v_{max}$  is a specified perimeter number. This constraint guarantees that the particle's travel stays inside a specific region. The size of the searching areas within the present and destination coordinates is determined by the value of  $v_{max}$ .

The basic PSO (Kennedy and Eberhart, 1995) performs well for different optimization problems. The basic algorithm constraints have been found through thorough experimentation with a standard collection of CEC benchmark functions (Cheng et al., 2018). By increasing solution diversity during initialization and algorithm runs, these modifications aim to address limited exploration. There were three basic PSO alterations.

The initial alteration incorporates chaotic elite learning to enhance individual particles by generating novel solutions nearby, which ensures population diversity and helps escape sub-optimal regions. A logistic map, as stated in Eq. (4), is the focus of this modification:

$$C_{i+1} = 4 \times C_i \times (1 - c_i) \quad (4)$$

in which  $i$  represents the integration number,  $c_i$  denotes a chaotic value in round  $i$ , while the initial value of  $c_0$  is synthesized randomly inside  $[0, 1]$ . The optimum individual  $X_i$  updates its location by utilizing Eq. (5):

$$X_{i,j}^1 = X_{i,j} + rand \times (c_i - 1) \quad (5)$$

where  $X_{i,j}^1$  represents the  $j$ th part of the best solution.

Further on, the algorithm employs the Lévy flight procedure to enhance updates of the positions. By providing each solution the opportunity arises to occasionally execute long flight jumps. This facilitates escaping from sub-optimal areas and improves the general performance of the individual particle. While the algorithm converges to the promising region, this long flight distance is dynamically decreased.

$$X_{i,j}^1 = X_{i,j}^{best} \times L \times e \quad (6)$$

$$e = \frac{\bar{a}}{2} \quad (7)$$

where  $X_{i,j}$  represent an individual's components that are modified,  $X_{i,j}^{best}$  denotes the best solution, while  $e$  corresponds to the scaling factor and  $\bar{a}$  is obtained with respect to the following:

$$\bar{a} = 1 + \cos(\pi + \frac{t}{T}) \quad (8)$$

where  $t$  marks the ongoing round, and  $T$  represents the maximum count of rounds. The value of  $L$  is the Lévy flight distribution which can be obtained by:

$$L = s \times \frac{\mu \times \phi}{|v|^{\frac{1}{\tau}}} \quad (9)$$

where  $\tau$  represents the Lévy flight index,  $S$  denotes a constant preset to value of 0.01 preventing extremely remote flights, while  $\mu$  and  $v$  are arbitrarily selected within  $[0, 1]$ . The  $\phi$  value may be obtained by:

$$\phi = \left( \frac{\Gamma(1 + \tau) \times \sin(\frac{\pi \times \tau}{2})}{\Gamma(\frac{1 + \tau}{2}) \times \tau \times 2^{\frac{\tau-1}{2}}} \right)^{\frac{1}{\tau}} \quad (10)$$

here, the  $\Gamma$  value is calculated by:

$$\Gamma(1 + \tau) = \int_0^{\infty} x^{\tau} e^{-x} dx \quad (11)$$

Lastly, the third modification incorporates a low-level hybrid mechanism with the firefly algorithm (FA) (Yang and Slowik, 2020). The

FA's search procedure given by Eq. (12) is merged with the modified metaheuristics to further improve the exploration phase.

$$X_i^{t+1} = X_i^t + \beta_0 \cdot e^{-\gamma R_{i,j}^2} (X_j^t - X_i^t) + \alpha^t (\kappa - 0.5) \quad (12)$$

where  $\alpha$  is the randomization parameter,  $\kappa$  is a pseudo-random value drawn from the Gaussian distribution. The parameter  $\gamma$  represents the properties of the environment,  $\beta_0$  denotes the maximal attractiveness of a single solution, and  $R_{i,j}$  marks the range among solutions.

To allow fair metaheuristics contributions, it was necessary to implement a supplementary search control variable  $\theta$ . If the value of  $\theta$  is larger than the current iteration  $t$ , every individual randomly selects which algorithm will be employed. A random number is generated from a uniform distribution within  $[0, 1]$ . If this value is larger than 0.5 the FA search procedure is employed. For this research, a  $\theta$  value was empirically established, and the best outcomes were obtained at  $\theta = 3$ .

The formulated metaheuristics were labeled PSO with dynamic strategy change (PSO-DSC). The pseudocode of the PSO-DSC is explained in Algorithm 1.

---

#### Algorithm 1 PSO-DSC pseudo-code

---

```

1: Initialize parameters
2: Synthesize a populace  $X$  by employing chaotic initialization
3: while ( $t < T$ ) do
4:   Calculate fitness of all solutions and note the best
5:   for ( each solution in  $X$  ) do
6:     if  $t > \theta$  then
7:       Utilize PSO search procedure
8:     else
9:       Produce random number  $rnd$ 
10:      if  $rnd < 0.5$  then
11:        Utilize PSO search procedure
12:      else
13:        Utilize FA search procedure
14:     $t = t + 1$ 
15: Return the best solution from  $X$ 

```

---

## 4. Experimental setup, analysis and discussion

The suggested methods are evaluated by employing real-world spot pricing data from Amazon EC2's 'us-west' area. The dataset covers huge Linux VMs. Inputs include 'c38xlarge', 'm44xlarge', and 'r38xlarge', with the final price forecasting one step ahead. The dataset analysis starts from 24.03.2017. Due to the huge dataset sample size, a smaller sub-sample is used. Kaggle hosts the original dataset.<sup>1</sup> The available data is a 10-lag time-series. Data is split into training validation and evaluation subsets for investigation. An initial 70% (1200 samples) are allotted for training, 10% for validation and the remainder for evaluation.

Several metaheuristic optimizers are applied to the optimization of both LSTM and GRU networks in two separate experiments. A comparative analysis is carried out between the introduced modified algorithm and the original PSO (Kennedy and Eberhart, 1995). Additional algorithms included in the comparisons are "genetic algorithm" (Mirjalili and Mirjalili, 2019) (GA) and COLSHADE (Gurrola-Ramos et al., 2020) optimizes. Well-established swarm optimizes such as the "firefly algorithm" (Yang, 2009) (FA), "artificial bee colony" (Karaboga and Basturk, 2008) (ABC) and "Harris hawk optimizes" (Heidari et al., 2019) (HHO) are also included. The "variable neighborhood search" (Mladenović and Hansen, 1997) (VNS) algorithm is also used in the comparative analysis. Each optimizer is given a population size of five and five

<sup>1</sup> <https://www.kaggle.com/datasets/berkayalan/aws-ec2-instance-comparison>

iterations to enhance population quality. To accommodate for randomness, simulations are conducted over 30 times independently. To prevent over-training an early stopping criterion is used based on the equal to one-third of training epochs. Parameter ranges from both LSTM and GRU are optimized from a constrained including learning rate  $[0, 0001, 0.01]$ , dropout  $[0.05, 0.2]$ , epochs  $[10, 30]$ , neurons per layer  $[10, 30]$ .

Evaluation of models involves standard regressing measures such as "root mean square error" (RMSE), "mean absolute error" (MAE), "mean square error" (MSE), and "coefficient of determination" ( $R^2$ ). The objective metric is MSE, and  $R^2$  is the indicator function. An additional metric, the index of agreement (IoA), was used to evaluate synthesized models (Eq. (13)).

$$IoA = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (|y_i - \bar{y}| + |y_i - \hat{y}_i|)^2}, \quad (13)$$

where  $y_i$  and  $\hat{y}_i$  one more time denote the real and predicted values and  $\bar{y}$  is the average value of real data. If this index has a significant value the model's performance is good, suggesting strong agreement.

A flowchart of the proposed experimental framework describing dataset preparation, and the optimization process is provided in Fig. 1.

### 4.1. Simulation outcomes

#### 4.1.1. Simulations with LSTM networks

The objective (MSE) function outputs for the best, worst mean, and median LSTM simulations are presented in Table 1, followed by comprehensive evaluations within the most promising models. The proposed optimizer demonstrates impressive outcomes, outperforming all competing metaheuristics demonstrating an MSE of  $1.33E-06$  in the best-case execution. Furthermore, the proposed optimizer achieved the highest outcome for the median and mean cases demonstrating an objective function score of  $1.48E-06$  and  $1.33E-06$  respectively. The FA algorithm demonstrates adorable stability, with more stability evaluations shown in Fig. 2. As showcased in the stability diagrams, the introduced algorithm outperforms the original algorithm in all cases as well as competing metaheuristics demonstrating better results. While the stability is comparatively lower than some optimizers such as the original PSO, GA, and FA, the results attained by these optimizers are below even the worst results attained by the introduced optimizer.

#### 4.1.2. Simulations with GRU networks

Outcomes in terms of objective (MSE) function for the best, worst mean and median GRU simulations followed by detailed comparisons between the best models are provided in Table 2. The introduced optimizer demonstrates impressive outcomes, outperforming all competing metaheuristics, producing an MSE of  $1.16E-06$  in the best-case execution, and outperforming LSTM models as well. Furthermore, the introduced optimizer attained the best outcome for the worst, median, and mean cases demonstrating an objective function score of respectively. The highest rate of stability is showcased by the original PSO algorithm despite this optimizer not matching the performance of the modified algorithm. Further comparisons in terms of stability are provided in Fig. 3. As showcased in the stability diagrams, the introduced algorithm outperforms the original algorithm in all cases as well as competing metaheuristics demonstrating better results. While the stability is comparatively lower than some optimizers such as the original PSO, GA, and FA, the results attained by these optimizers are below even the worst results attained by the introduced optimizer. The best-performing model was optimized by the introduced modified metaheuristic and demonstrates the best outcomes across all metrics for the given challenge.

Further details on optimizer capabilities to balance exploration and exploitation while avoiding local minimum traps are provided in Fig. 4. The introduced optimizer manages to avoid a local trap in the third iteration of the optimization locating a more promising area of

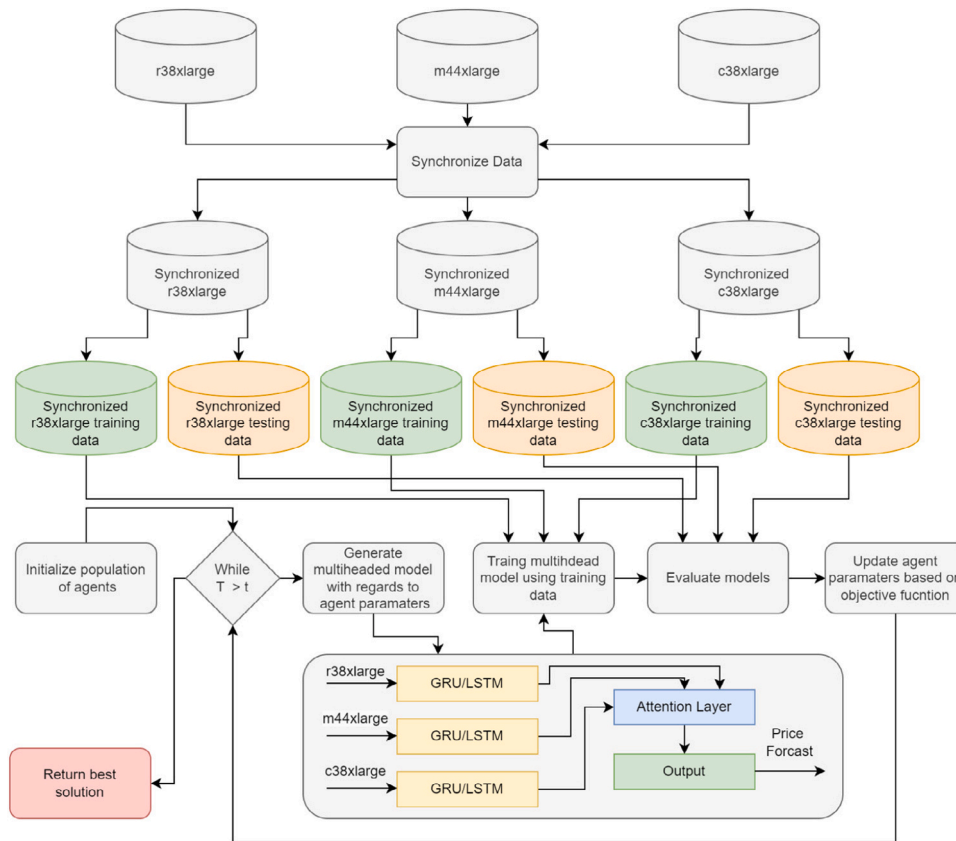


Fig. 1. Introduced optimization framework flowchart.

Table 1  
Outcomes for LSTM simulations.

Overall comparisons						
Method	Best	Worst	Mean	Median	Std	Var
MHLSTM-PSO-DSC	<b>1.33E-06</b>	1.66E-06	<b>1.48E-06</b>	<b>1.33E-06</b>	1.67E-07	2.77E-14
MHLSTM-PSO	1.54E-06	1.64E-06	1.59E-06	1.59E-06	5.03E-08	2.53E-15
MHLSTM-GA	1.53E-06	<b>1.62E-06</b>	1.57E-06	1.53E-06	4.51E-08	2.04E-15
MHLSTM-VNS	1.71E-06	1.74E-06	1.72E-06	1.71E-06	1.44E-08	2.08E-16
MHLSTM-ABC	1.67E-06	1.78E-06	1.73E-06	1.78E-06	5.55E-08	3.08E-15
MHLSTM-FA	1.69E-06	1.70E-06	1.69E-06	1.70E-06	<b>2.17E-09</b>	<b>4.71E-18</b>
MHLSTM-HHO	1.41E-06	1.68E-06	1.53E-06	1.41E-06	1.35E-07	1.81E-14
MHLSTM-COLSHADE	1.67E-06	1.89E-06	1.79E-06	1.89E-06	1.07E-07	1.15E-14
Detailed Comparisons						
Method	R <sup>2</sup>	MAE	MSE	RMSE	IoA	
MHLSTM-PSO-DSC	<b>0.191541</b>	<b>0.000908</b>	<b>1.33E-06</b>	<b>0.001152</b>	<b>0.531649</b>	
MHLSTM-PSO	0.061297	0.000990	1.54E-06	0.001241	0.510089	
MHLSTM-GA	0.066289	0.001044	1.53E-06	0.001238	0.472012	
MHLSTM-VNS	-0.043414	0.001118	1.71E-06	0.001308	0.142093	
MHLSTM-ABC	-0.020013	0.001119	1.67E-06	0.001294	0.314240	
MHLSTM-FA	-0.030752	0.001149	1.69E-06	0.001300	0.200151	
MHLSTM-HHO	0.139422	0.000971	1.41E-06	0.001188	0.433447	
MHLSTM-COLSHADE	-0.018482	0.001127	1.67E-06	0.001293	0.115080	

the search space by the 3rd iteration, suggesting that the introduced modifications help the modified algorithm locate more promising areas within the search space.

Parameter selections for the best GRU models as follows: a learning rate of 0.000194, dropout of 0.050000, 30 training epochs neurons in the first head 11, 30 in the second and 14 in the third heads. A total of 28 is utilized in the dense concatenation layer.

#### 4.2. Comparison between the best LSTM and GRU models

Both LSTM and GRU-optimized models have similar objective and indicator functions, with the modified PSO-DSC algorithm optimizing the best models. A comparison between the best performing GRU, LSTM, simple feed-forward artificial neural networks (ANN), and other contemporary regression forecasting models such as extreme gradient boosting (XGBoost), Support Vector Machines (SVM), decision trees and

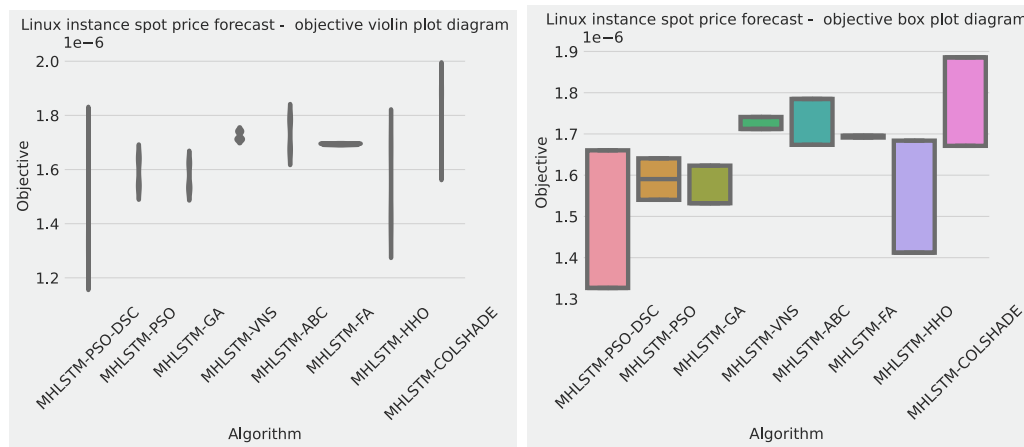


Fig. 2. Objective function distribution plots for LSTM optimizations.

Table 2  
Outcomes for each optimizer included in comparative analysis.

Overall Comparisons						
Method	Best	Worst	Mean	Median	Std	Var
MHGRU-PSO-DSC	1.16E-06	1.23E-06	1.20E-06	1.23E-06	3.41E-08	1.16E-15
MHGRU-PSO	1.37E-06	1.38E-06	1.38E-06	1.37E-06	3.86E-09	1.49E-17
MHGRU-GA	1.64E-06	2.05E-06	1.90E-06	2.05E-06	1.97E-07	3.89E-14
MHGRU-VNS	1.32E-06	1.33E-06	1.32E-06	1.33E-06	5.86E-09	3.44E-17
MHGRU-ABC	1.41E-06	1.63E-06	1.52E-06	1.52E-06	1.09E-07	1.20E-14
MHGRU-FA	1.47E-06	1.48E-06	1.47E-06	1.47E-06	5.92E-09	3.51E-17
MHGRU-HHO	1.30E-06	1.62E-06	1.47E-06	1.62E-06	1.63E-07	2.65E-14
MHGRU-COLSHADE	1.69E-06	1.96E-06	1.81E-06	1.69E-06	1.32E-07	1.76E-14
Detailed Comparisons						
Method	R <sup>2</sup>	MAE	MSE	RMSE	IoA	
MHGRU-PSO-DSC	0.290191	0.000801	1.16E-06	0.001079	0.678492	
MHGRU-PSO	0.162884	0.000985	1.37E-06	0.001172	0.389012	
MHGRU-GA	0.000602	0.001105	1.64E-06	0.001281	0.056272	
MHGRU-VNS	0.197970	0.000963	1.32E-06	0.001147	0.592289	
MHGRU-ABC	0.139642	0.000987	1.41E-06	0.001188	0.435323	
MHGRU-FA	0.107040	0.001065	1.47E-06	0.001210	0.435001	
MHGRU-HHO	0.209853	0.000944	1.30E-06	0.001139	0.560155	
MHGRU-COLSHADE	-0.030043	0.001084	1.69E-06	0.001300	0.179363	

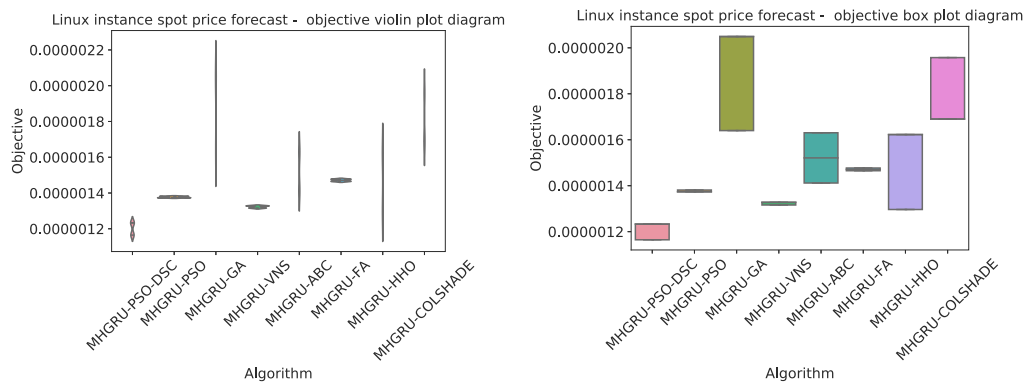


Fig. 3. Objective function distribution plots for GRU optimizations.

random forest models, as well as statistical forecasting techniques, such as autoregressive integrated moving average (AIRMA), is provided in Table 3 in terms of detailed metrics. The best-performing GRU model outperforms the best-performing LSTM across all metrics, confirming that the simple GRU architecture is better at forecasting cloud instance prices. This challenge benefits from the lower computational costs of GRU models in comparison to LSTM.

#### 4.3. Outcomes statistical validation

Methods with randomness, repeatability, and reliability must be evaluated. Parametric tests have to meet independence, normality, and homoscedasticity (LaTorre et al., 2021) criteria to ensure safe utilization. By using independent random seeds the independence criterion is fulfilled. The homoscedasticity criterion was evaluated through the

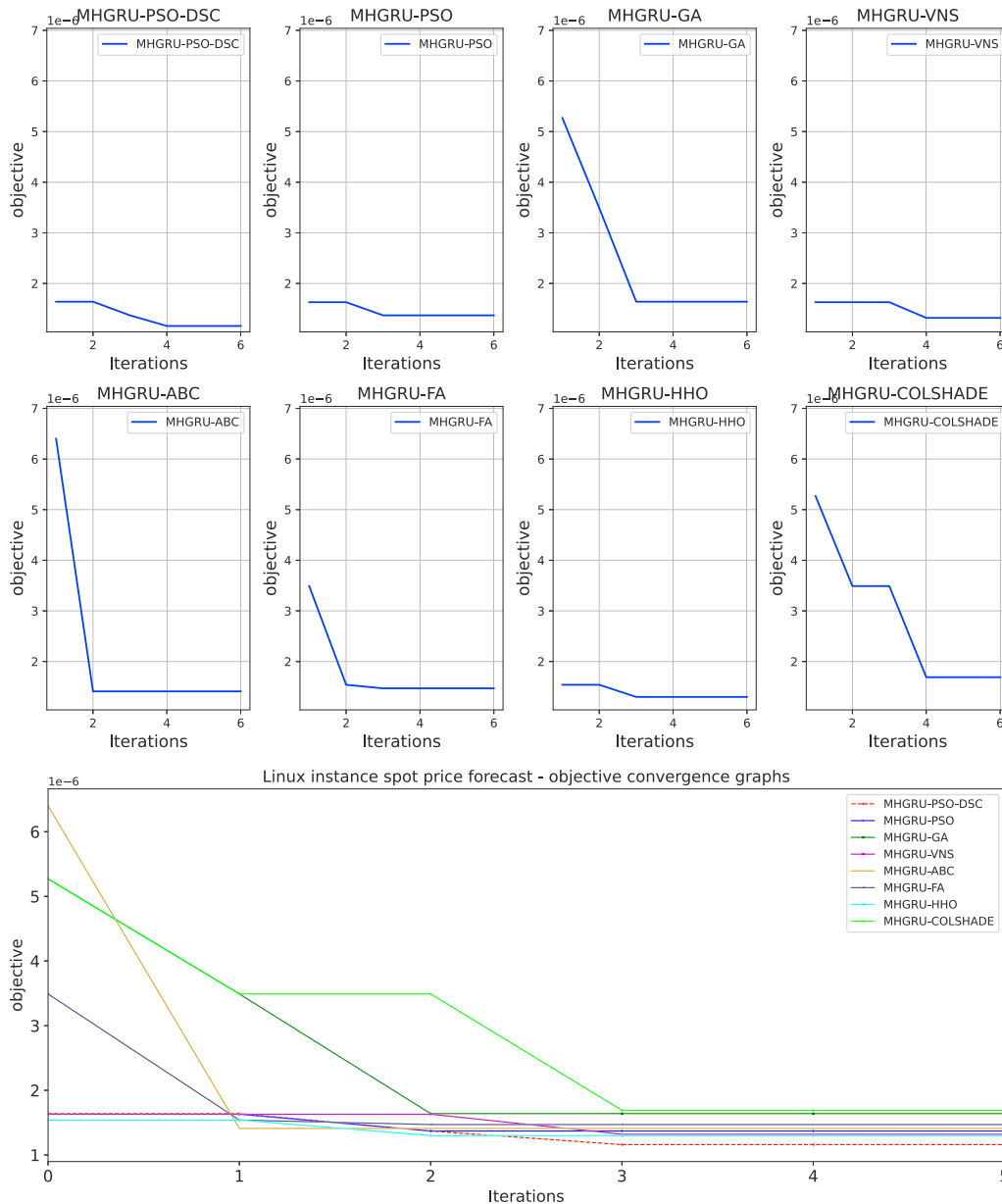


Fig. 4. Convergence diagrams for GRU optimizations.

**Table 3**  
Comparison between the best performing GRU and LSTM model.

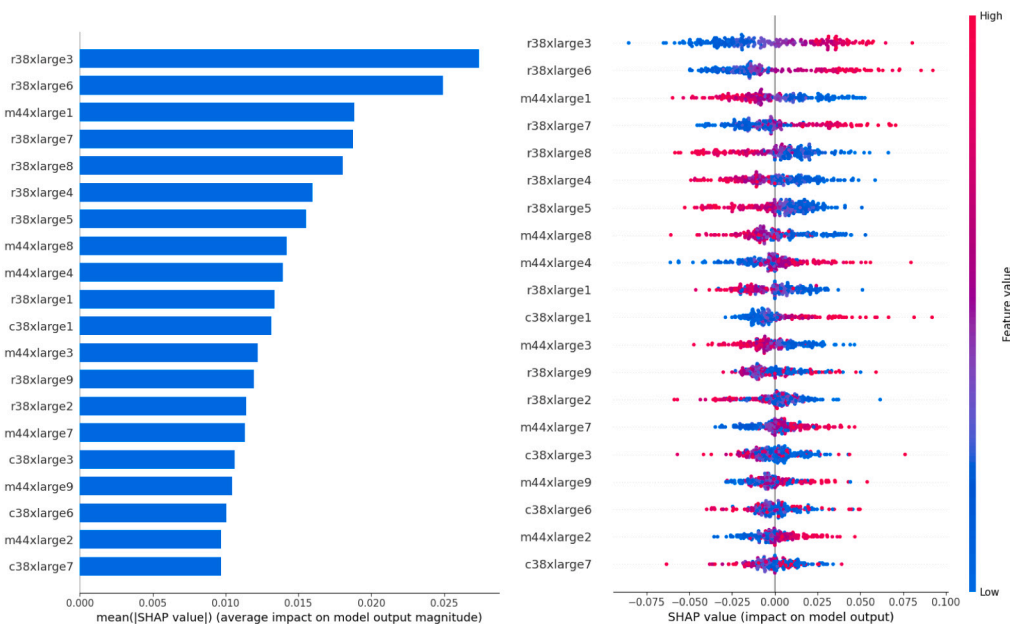
Method	R <sup>2</sup>	MAE	MSE	RMSE	IoA
MHGRU-PSO-DSC	<b>0.290191</b>	<b>0.000801</b>	<b>1.16E-06</b>	<b>0.001079</b>	<b>0.678492</b>
MHLSTM-PSO-DSC	0.191541	0.000908	1.33E-06	0.001152	0.531649
Linear Regression	-1.933209	0.001710	4.68E-06	0.002163	0.446984
Decision Tree Regression	-3.268749	0.002200	6.81E-06	0.002609	0.393007
Random Forest Regression	-1.049327	0.0015143	3.27E-06	0.001808	0.435379
XGBoost	-0.989057	0.001508	3.17E-06	0.001781	0.444499
SVR	-0.048330	0.001143	1.67E-06	0.001293	0.264830
Feedforward ANN	-4.708735	0.002499	9.10E-06	0.003017	0.367674
ARIMA	-7.810849	0.003529	1.40E-05	0.003748	0.000000

application of “Levene’s test” (Glass, 1966), which yielded the  $p$ -value of 0.68 for each case, therefore this condition is also met. To confirm normality Shapiro–Wilk’s single problem analysis is utilized (Shapiro and Francia, 1972) with outcomes shown in Table 4. Shapiro–Wilk’s  $p$ -values were determined separately for each regarded approach all

obtained  $p$ -values are below 0.05 and  $H_0$  is rejected. Since the conditions for parametric tests were not met, validation is conducted via non-parametric tests. The “Wilcoxon signed-rank” test was performed with the objective function outcomes (MSE). The outcomes of this test are provided in Table 4. As the  $p$ -values are in each case below threshold limit  $\alpha = 0.05$ , it means that the outcomes attained by the

**Table 4**  
Statistical evaluation outcomes.

Shapiro–Wilk outcomes for PSO-DSC								
Approach	PSO-DSC	PSO	GA	VNS	ABC	FA	HHO	COLSHADE
MHGRU	0.024	0.028	0.028	0.033	0.035	0.030	0.028	0.040
MHLSTM	0.023	0.027	0.034	0.030	0.028	0.030	0.030	0.041
Wilcoxon signed-rank test outcomes for MHLSTM simulations								
PSO-DSC vs. others	PSO	GA	VNS	ABC	FA	HHO	COLSHADE	
MHGRU	0.033	0.023	0.021	0.023	0.021	0.028	0.040	
MHLSTM	0.037	0.022	0.020	0.023	0.022	0.032	0.040	



**Fig. 5.** Best Performing GRU model SHAP interpretations.

proposed PSO-DSC approach are statistically significantly superior to other methods regarded in comparative analysis.

#### 4.4. Best-performing model interpretation

Understanding the factors that influence a model’s decisions is crucial. SHAP (Lundberg and Lee, 2017) is useful for obtaining insights. The best-constructed forecasting GRU model is subjected to SHAP interpretation. A high importance is given to r38xlarge samples 3 lags back followed by the same feature six months back. This is then followed by the m44xlarge one lag back. While their contribution is modest, the cumulative contributions contribute towards more accurate forecasting being made by the best-constructed model (see Fig. 5).

### 5. Conclusions

This study deals with the challenging and costly problem of cloud instance price forecasting. With more infrastructure relying on cloud services, a stable and predictable cost model will help companies supervise spending and optimize operations. Forecasting prices can help providers maintain consistency in billing clients. Multi-headed recurrent architecture is explored for LSTM and GRUs. Metaheuristic optimizers tune model parameters because network performance hinders architectural selection and training parameter sets. A modified version of the PSO optimizer enhances GRU and LSTM models, which forecast cloud instance pricing on a real-world dataset. The GRU models optimized by the new approach performed best with an MAE score of 0.000801. The proposed optimizer’s improvements enhance convergence rates, prevent premature stagnation and allow agents to find

better search space solutions. The outcomes are analyzed statistically, and the best model is analyzed using explainable AI approaches to determine the features that boost forecasts.

High computation demands of optimization and model training limit the dataset size for analysis. Furthermore, high processing demands limit the number of optimizers in the comparative analysis. Scalability concerns when applying the proposed methodology to a larger dataset might increase forecasting model training demands. It is intended to explore and refine the forecasting method in the future. As more resources become available, other well-established optimizers will be compared to the introduced approach, and its uses will be examined.

#### CRediT authorship contribution statement

**Mohamed Salb:** Writing – review & editing, Validation, Software, Methodology. **Luka Jovanovic:** Writing – review & editing, Visualization, Validation, Supervision, Software, Methodology, Data curation. **Ali Elsadai:** Writing – review & editing, Validation, Resources, Methodology, Funding acquisition, Formal analysis. **Nebojsa Bacanin:** Writing – review & editing, Writing – original draft, Validation, Supervision, Methodology. **Vladimir Simic:** Writing – review & editing. **Dragan Pamucar:** Writing – review & editing. **Miodrag Zivkovic:** Writing – review & editing, Validation, Supervision.

#### Declaration of competing interest

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

## Data availability

This study's dataset is publicly available from this Kaggle URL: <https://www.kaggle.com/datasets/berkayalan/aws-ec2-instance-comp-arison>.

## References

- Agmon Ben-Yehuda, O., Ben-Yehuda, M., Schuster, A., Tsafirir, D., 2013. Deconstructing Amazon EC2 spot instance pricing. *ACM Trans. Econ. Comput. (TEAC)* 1 (3), 1–20.
- Bacanin, N., Stoean, C., Zivkovic, M., Jovanovic, D., Antonijevic, M., Mladenovic, D., 2022. Multi-swarm algorithm for extreme learning machine optimization. *Sensors* 22 (11), 4204.
- Bacanin, N., Venkatachalam, K., Bezdan, T., Zivkovic, M., Abouhawwash, M., 2023. A novel fire algorithm approach for efficient feature selection with COVID-19 dataset. *Microprocess. Microsyst.* 98, 104778.
- Bu, S.J., Cho, S.B., 2020. Time series forecasting with multi-headed attention-based deep learning for residential energy consumption. *Energies* 13 (18), 4722.
- Cheng, R., Li, M., Tian, Y., Xiang, X., Zhang, X., Yang, S., Jin, Y., Yao, X., 2018. Benchmark Functions for the CEC'2018 Competition on Many-Objective Optimization. Technical Report.
- Dadashov, E., Cetintemel, U., Kraska, T., 2014. Putting analytics on the spot: or how to lower the cost for analytics. *IEEE Internet Comput.* 18 (5), 70–73.
- Darbandi, M., 2017a. Kalman filtering for estimation and prediction servers with lower traffic loads for transferring high-level processes in cloud computing. Published by HCTL Int. J. Technol. Innov. Res. (ISSN: 2321-1814) 23 (1), 10–20.
- Darbandi, M., 2017b. Proposing new intelligence algorithm for suggesting better services to cloud users based on Kalman filtering. Published by J. Comput. Sci. Appl. (ISSN: 2328-7268) 5 (1), 11–16.
- Darbandi, M., Abedi, M., Fard, S., Nakhodchi, S., 2012. Involving Kalman filter technique for increasing the reliability and efficiency of cloud computing. In: Proceedings of the International Conference on Scientific Computing. CSC, The Steering Committee of The World Congress in Computer Science, Computer ... , p. 1.
- Darbandi, M., Haghgoo, S., Hajiali, M., Khabir, A., 2018. Prediction and estimation of next demands of cloud users based on their comments in CRM and previous usages. In: 2018 International Conference on Communication, Computing and Internet of Things. IC3IoT, IEEE, pp. 81–86.
- Dey, R., Salem, F.M., 2017. Gate-variants of gated recurrent unit (GRU) neural networks. In: 2017 IEEE 60th International Midwest Symposium on Circuits and Systems. MWSCAS, IEEE, pp. 1597–1600.
- Glass, G.V., 1966. Testing homogeneity of variances. *Am. Educ. Res. J.* 3 (3), 187–190.
- Greff, K., Srivastava, R.K., Koutník, J., Steunebrink, B.R., Schmidhuber, J., 2016. LSTM: A search space odyssey. *IEEE Trans. Neural Netw. Learn. Syst.* 28 (10), 2222–2232.
- Gurrola-Ramos, J., Hernández-Aguirre, A., Dalmau-Cedeño, O., 2020. COLSHADE for real-world single-objective constrained optimization problems. In: 2020 IEEE Congress on Evolutionary Computation. CEC, IEEE, pp. 1–8.
- Heidari, A., Jafari Navimipour, N., 2022. Service discovery mechanisms in cloud computing: a comprehensive and systematic literature review. *Kybernetes* 51 (3), 952–981.
- Heidari, A., Jafari Navimipour, N., Unal, M., Zhang, G., 2023. Machine learning applications in internet-of-drones: Systematic review, recent deployments, and open issues. *ACM Comput. Surv.* 55 (12), 1–45.
- Heidari, A.A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., Chen, H., 2019. Harris hawks optimization: Algorithm and applications. *Future Gener. Comput. Syst.* 97, 849–872.
- Heidari, A., Navimipour, N.J., Otsuki, A., 2024a. Cloud-based non-destructive characterization. *Non-Destruct. Mater. Charact. Methods* 727–765.
- Heidari, A., Shishehlou, H., Darbandi, M., Navimipour, N.J., Yalcin, S., 2024b. A reliable method for data aggregation on the industrial internet of things using a hybrid optimization algorithm and density correlation degree. *Cluster Comput.* 1–19.
- Jovanovic, D., Antonijevic, M., Stankovic, M., Zivkovic, M., Tanaskovic, M., Bacanin, N., 2022. Tuning machine learning models using a group search firefly algorithm for credit card fraud detection. *Mathematics* 10 (13), 2272.
- Jovanovic, L., Antonijevic, M., Zivkovic, M., Jovanovic, D., Marjanovic, M., Bacanin, N., 2022a. Sine cosine algorithm with tangent search for neural networks dropout regularization. In: *Data Intelligence and Cognitive Informatics: Proceedings of ICDICI 2022*. Springer, pp. 789–802.
- Jovanovic, L., Zivkovic, M., Antonijevic, M., Jovanovic, D., Ivanovic, M., Jassim, H., 2022b. An emperor penguin optimizer application for medical diagnostics. In: 2022 IEEE Zooming Innovation in Consumer Technologies Conference. ZINC, IEEE, pp. 191–196.
- Karaboga, D., Basturk, B., 2008. On the performance of artificial bee colony (ABC) algorithm. *Appl. Soft Comput.* 8 (1), 687–697.
- Kennedy, J., Eberhart, R., 1995. Particle swarm optimization. In: *Proceedings of ICNN'95-International Conference on Neural Networks*. 4, IEEE, pp. 1942–1948.
- LaTorre, A., Molina, D., Osaba, E., Poyatos, J., Del Ser, J., Herrera, F., 2021. A prescription of methodological guidelines for comparing bio-inspired optimization algorithms. *Swarm Evol. Comput.* 67, 100973.
- Lundberg, S.M., Lee, S.L., 2017. A unified approach to interpreting model predictions. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (Eds.), *Advances in Neural Information Processing Systems* 30. Curran Associates, Inc., pp. 4765–4774.
- Mell, P., Grance, T., et al., 2011. The NIST definition of cloud computing.
- Mirjalili, S., Mirjalili, S., 2019. Genetic algorithm. *Evol. Algorithms Neural Netw. Theory Appl.* 43–55.
- Mladenović, N., Hansen, P., 1997. Variable neighborhood search. *Comput. Oper. Res.* 24 (11), 1097–1100.
- Norozpour, S., Darbandi, M., 2020. Proposing new method for clustering and optimizing energy consumption in WSN. *Talent Dev. Excell.* 12.
- Shapiro, S.S., Francia, R., 1972. An approximate analysis of variance test for normality. *J. Am. Stat. Assoc.* 67 (337), 215–216.
- Singh, V.K., Dutta, K., 2015. Dynamic price prediction for amazon spot instances. In: 2015 48th Hawaii International Conference on System Sciences. IEEE, pp. 1513–1520.
- Wolpert, D.H., Macready, W.G., 1997. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* 1 (1), 67–82.
- Yang, X.S., 2009. Firefly algorithms for multimodal optimization. In: *Stochastic Algorithms: Foundations and Applications: 5th International Symposium, SAGA 2009, Sapporo, Japan, October 26–28, 2009*. Proceedings 5. Springer, pp. 169–178.
- Yang, X.S., Slowik, A., 2020. Firefly algorithm. In: *Swarm Intelligence Algorithms*. CRC Press, pp. 163–174.
- Yazdi, A.K., Komasi, H., 2024. Best practice performance of COVID-19 in America continent with artificial intelligence. *Spectr. Oper. Res.* 1 (1), 1–12. <http://dx.doi.org/10.31181/sor1120241>.
- Younas, R., Ul Haq, H.B., Baig, M.D., 2024. A framework for extensive content-based image retrieval system incorporating relevance feedback and query suggestion. *Spectr. Oper. Res.* 1 (1), 13–32.
- Zivkovic, M., Jovanovic, L., Ivanovic, M., Krdzic, A., Bacanin, N., Strumberger, I., 2022. Feature selection using modified sine cosine algorithm with COVID-19 dataset. In: *Evolutionary Computing and Mobile Sustainable Networks: Proceedings of ICECMNS 2021*. Springer, pp. 15–31.