

## Research Article

# A New ANN-Particle Swarm Optimization with Center of Gravity (ANN-PSOCoG) Prediction Model for the Stock Market under the Effect of COVID-19

Razan Jamous,<sup>1</sup> Hosam ALRahhal <sup>1,2</sup> and Mohamed El-Darieby<sup>1</sup>

<sup>1</sup>Faculty of Engineering and Applied Science, University of Regina, Regina, SK S4S 0A2, Canada

<sup>2</sup>Faculty of Engineering, Nahda University, Beni Suef 62764, Egypt

Correspondence should be addressed to Hosam ALRahhal; [hosamrahhal@gmail.com](mailto:hosamrahhal@gmail.com)

Received 22 October 2020; Revised 12 November 2020; Accepted 12 April 2021; Published 30 April 2021

Academic Editor: Ligang He

Copyright © 2021 Razan Jamous et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Since the declaration of COVID-19 as a pandemic, the world stock markets have suffered huge losses prompting investors to limit or avoid these losses. The stock market was one of the businesses that were affected the most. At the same time, artificial neural networks (ANNs) have already been used for the prediction of the closing prices in stock markets. However, standalone ANN has several limitations, resulting in the lower accuracy of the prediction results. Such limitation is resolved using hybrid models. Therefore, a combination of artificial intelligence networks and particle swarm optimization for efficient stock market prediction was reported in the literature. This method predicted the closing prices of the shares traded on the stock market, allowing for the largest profit with the minimum risk. Nevertheless, the results were not that satisfactory. In order to achieve prediction with a high degree of accuracy in a short time, a new improved method called PSOCoG has been proposed in this paper. To design the neural network to minimize processing time and search time and maximize the accuracy of prediction, it is necessary to identify hyperparameter values with precision. PSOCoG has been employed to select the best hyperparameters in order to construct the best neural network. The created network was able to predict the closing price with high accuracy, and the proposed model ANN-PSOCoG showed that it could predict closing price values with an infinitesimal error, outperforming existing models in terms of error ratio and processing time. Using S&P 500 dataset, ANN-PSOCoG outperformed ANN-SPSO in terms of prediction accuracy by approximately 13%, SPSOCOG by approximately 17%, SPSO by approximately 20%, and ANN by approximately 25%. While using DJIA dataset, ANN-PSOCoG outperformed ANN-SPSO in terms of prediction accuracy by approximately 18%, SPSOCOG by approximately 24%, SPSO by approximately 33%, and ANN by approximately 42%. Besides, the proposed model is evaluated under the effect of COVID-19. The results proved the ability of the proposed model to predict the closing price with high accuracy where the values of MAPE, MAE, and RE were very small for S&P 500, GOLD, NASDAQ-100, and CANUSD datasets.

## 1. Introduction

At the beginning of 2020, the world witnessed a large spread of the COVID-19 virus, which was discovered in the Chinese city of Wuhan and was classified as a pandemic by the World Health Organization on March 11, 2020 [1].

Due to the spread of the COVID-19 pandemic, unprecedented measures have been taken to slow down this virus's spread. Hundreds of millions of people were asked to stay inside their homes. Air, sea, and land transport stopped, and many factories were suspended. As a result, stock

markets were among the worst affected by the COVID-19 pandemic, which led to a large economic crisis, where the US Gross Domestic Product of 2020 was affected by significant short-term hit by 1.5% [2], while in China, it was between 2% and 3% [3]. The COVID-19 pandemic affects the stock markets through feelings of fear and panic affecting investors and resulting from the suspension of many economic sectors such as industry, transport, and tourism, as well as the lack of hope in ending the pandemic in the close future [4]. The stock exchange is a preferred choice for investors because it brings them high profits. However, forecasting in

the stock exchange is a complex process because stock prices depend on several factors and are greatly affected by rumors [5]. The stock markets attracted researchers' attention to analyze and predict future values to guide the investors to make correct buying or selling decisions and achieve the best profit with the minimum risk.

Due to the significant need for price prediction in the stock market, many methods were presented to achieve this aim. There is also a growing interest to employ AI techniques to forecast the prices in the stock market. One of the most preferred and common in use methods is ANN that can be used purely or combined with other statistical methods [6]. In some cases of complex nonlinear systems like stock markets, ANN fails to learn the actual processes associated with the corresponding target variable. Also, the ANN model is found to be efficient in some cases and the same ANN model may be inefficient in other cases.

Many researchers have been studying stock price prediction for years; some of these studies have improved the existing models, and some have further processed the data. These studies are not perfect, some of the models are too complex, and some of the processing procedures are time-consuming. These shortcomings will increase the methods' instability and limit the application and extension of the research results.

The traditional stock forecasting methods could not fit and analyze the highly nonlinear and multifactor stock market well, so there are low prediction accuracy and slow training speed problems [7]. Therefore, recent studies such as fusion in stock market prediction [6], financial trading strategy system [8], and short-term stock market price trend predictions [9] try to find solutions to problems. Some of these problems are (1) inaccurate prediction and (2) taking a long time as a result of incorrectly selecting the hyperparameters values of the neural network. When using artificial neural networks in forecasting, the selection of hyperparameter values is crucial to construct the artificial neural network's best topology to achieve high accuracy prediction of prices with the minimum possible error.

Hyperparameters are defined as the parameters that should be tuned before the beginning of the ANN training process. The hyperparameters for ANN technique include, for example, the number of hidden layers (NHL), number of neurons per hidden layer (NNPHL), activation function (ACTFUN) type, and learning rate (LR). These hyperparameters have the highest priority in designing an ANN and have a massive impact on ANN's performance. Consequently, multilayers ANN (MLANN) term brings first to mind multilayered and multinoded complex architectures. Learning of ANN is also thought to have taken its power from this complexity. NHL and NNPHL are variables which determine the network structure. They are the most mysterious hyperparameters that increase the number of parameters to be trained with. Since NHL and NNPHL are the quantities that affect the number of parameters to be trained (weight, bias), the most optimal model, i.e., "the most complex model with least parameters," must be established in both time and learning [10, 11]. Increasing or decreasing the number of hidden layers or the number of hidden nodes

might improve the accuracy or might not. This depends on the complexity of the problem, where inappropriate selection of these hyperparameters causes underfitting or overfitting problems [12]. For example, ANNs are universal function approximators and for them to learn to approximate a function or to predict a task, they need to have enough 'capacity' to learn the function.

NNPHL is the main measure of the learning model capacity. For a simple function, it might need a fewer number of hidden units. The more complex the function, the more learning capacity the model will need. Increasing the number of units by small number is not a problem, but a much larger number will lead to overfitting; for instance, providing a model with too much capacity, it might tend to overfitting problem, trying to "memorize" the dataset affecting the capacity to generalize [13]. As seen, it is very important to select the best values of these hyperparameters.

In neural networks (NNs), activation functions are used to transfer the values from one layer to another. The motivation for the activation functions defined by multilayer neural networks is to compose simple transformations to obtain highly nonlinear ones. In particular, MLPs compose affine transformations and elementwise nonlinearities. With the appropriate choice of parameters, multilayer neural networks can, in principle, approximate any smooth function with more hidden units allowing one to achieve better approximations. Also, activation functions are applied after every layer and need to be calculated millions of times in some cases. Hence, they are computationally inexpensive to calculate [12]. An inappropriate selection of the activation function can lead to the loss of information of the input during forwarding propagation and the exponential vanishing/exploding of gradients during backpropagation, or it can lead to high computational expenses [14].

Learning rate is one of the most important hyperparameters to tune ANN to achieve better performance. Learning rate determines the step size at each training iteration while moving towards an optimum of a loss function. The amount that the weights and bias parameters are updated is known as the learning rate. If the learning rate (LR) model is way smaller than the optimal value, it leads the model to learn slowly, up to hundreds or thousands of epochs, to reach an ideal state. LR that is too small may get stuck in an undesirable local minimum, and overfitting may occur. On the other hand, if the learning rate is much larger than the optimal value, it leads the model to learn much faster, but it would overshoot the ideal state, and the algorithm might not converge. This might cause undesirable divergent behavior in loss function [15].

The determination of the best hyperparameters depends on selecting suitable parameters that have to be tuned. The art of the network's hyperparameters optimization amounts to ending up at the balance point between underfitting and overfitting. After implementing the optimal value for each hyperparameter in an iterative way, the optimal hyperparameters will build the best architecture for ANN. This new model can improve the accuracy of future forecasted data and reduce the prediction process's consuming time. However, this procedure remains a challenging task.

Choosing inappropriate values for these variables leads to building an inappropriate network structure and gives inaccurate results that require many iterations to improve performance, which consumes a lot of time. These shortages lead to the need for hybridization. The hybrid model integrates two or more ANN and other models to overcome certain standalone ANN models' limitations and, hence, increase prediction accuracy and reduce time. The hybrid model is used for modeling complex target variables that involve different processes. Generally, the hybrid model consists of a decision-making integrated with a learning model. The decision-making model helps ANN to select the best hyperparameters values. For that, various techniques can be used as decision-making models such as particle swarm optimization (PSO) [16], backpropagation [17], genetic algorithms [18], ant colony optimization [19], bee swarm optimization (BSO) [20], Tabu search [21], and fuzzy systems [22].

As stated before, ANNs have already been employed for the prediction of closing price in stock markets. But standalone ANN has several limitations, resulting in the lower accuracy of the prediction results. However, the limitation of ANN is resolved using hybrid models.

In this paper, using a physical concept called the center of gravity (CoG), an enhanced PSO algorithm called PSO with the center of gravity (PSOCoG) is proposed. PSOCoG was used to select the best values of ANN's hyperparameters; the resulting model is called ANN-PSOCoG. The proposed ANN-PSOCoG model was used to forecast the closing price in the stock market. The effect of COVID-19 on ANN-PSOCoG was also studied. In the proposed ANN-PSOCoG hybrid model, the new enhanced algorithm (PSOCoG) is used as a decision-making model, and ANN is used as a learning model.

The essential contributions of this paper can be summarized as follows:

- (1) We design a generic model that can be used to select the best values of hyperparameters for an ANN
- (2) We find the best configuration for ANN in terms of NHL, NNPHL, ACTFUN, and LR using the proposed PSOCoG for ANN
- (3) The proposed model can be utilized to estimate the closing price using different historical data of stock market indices with high accuracy and minimum error
- (4) We compare the proposed model with traditional ANN, the standard PSO (SPSO), and ANN with standard PSO models
- (5) We study the efficiency of the proposed model under the effect of COVID-19

The rest of the paper is organized as follows. Section 2 introduces the background of standard particle swarm optimization and the stock market. A literature review is summarized in Section 3. The description of the proposed technique is given in Section 4. The performance evaluation is presented in Section 5, followed by a conclusion and future work in Section 6.

## 2. Background

For the paper to be self-content, throughout the following subsections, a brief review on ANN, SPSO, and stock market is introduced.

*2.1. Artificial Neural Networks (ANNs).* A classical ANN consists of three related layers: the input layer, the hidden layer, and the output layer. The number of units in the input and output layers depends on the input and output data's size. While the input units receive original data from the outside world and provide it to the ANN, no processing is executed in any of the input units. Instead, these units pass on information to the hidden units. The hidden nodes perform data processing and transfer information from the input units to the output units. The information is then processed and transmitted by output units from the ANN to the outside world.

Figure 1 shows a model for artificial neuron. In this model, there are various inputs to the neural network  $x_0, x_1, x_2, \dots, x_n$  which represent one single observation. Let  $W_{ij} = (W_{1j}, W_{2j}, \dots, W_{nj})$  be a weight vector of the  $j$ <sub>th</sub> unit of the hidden layer and  $b$  be a bias value that permits moving the transfer function up or down; the output is created through an activation function by the summation of multiplication of each input with the related weight vector. Mathematically, this can be expressed in the following equation:

$$x_1 \cdot W_{1j} + x_1 \cdot W_{2j} + x_1 \cdot W_{3j} + \dots + x_1 \cdot W_{nj} = \sum x_i \cdot W_{ij}. \quad (1)$$

After the output is produced, an activation function ( $\varphi \sum x_i \cdot W_{ij}$ ) is used to transmit the information to the outside world. The activation functions are used to get the output of the node. It is used to determine the output of ANN such as yes or no. The activation functions map the resulting values to a range of values between 0 and 1 or -1 to 1, and so on, depending upon the type of function used.

The activation functions can be basically divided into linear activation function and nonlinear activation functions. There are many types of activation functions such as log-sigmoid (logsig), softmax', tan-sigmoid (tansig), and the linear activation function (purelin). The most commonly used activation functions for multilayer networks are log-sigmoid (logsig), softmax', tan-sigmoid (tansig), and the linear activation function (purelin) [23].

ANN systems work in a different way, where the input data are combined to predict the output data. These systems are optimized in a way to allow for the computation of errors. To compute errors in these systems, the expected ANN's outputs are compared to the real targets. These systems have iteration ability where errors can be computed and estimated again, leading to decrease in errors, until the smallest possible error is found.

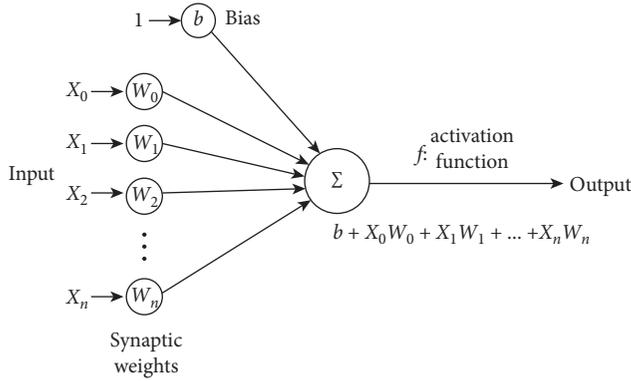


FIGURE 1: Artificial neuron model.

**2.2. Standard Particle Swarm Optimization (SPSO).** SPSO was presented by Kennedy and Eberhart [24]. SPSO mimics the simple behavior of organisms and the local cooperation with the environment and neighbor's organisms to develop behaviors used for solving complex problems, such as optimization issues. PSO algorithm has many advantages compared to different swarm intelligence (SI) techniques. For instance, its search procedure is simple, effective, and easy to implement. It can effectively find the best global solutions with high accuracy. PSO is a population-based search procedure in which each individual procedure represents a particle, i.e., a possible solution. These particles are grouped into a swarm. The particles moving within a multidimensional space adapt their locations depending on their experience and neighbors. The principle of the PSO technique can be explained [25] as follows. Let  $X_i(t) = (x_{i1}^{(t)}, x_{i2}^{(t)}, \dots, x_{id}^{(t)})$  denote the position particle<sub>*i*</sub> and  $V_i(t) = (v_{i1}^{(t)}, v_{i2}^{(t)}, \dots, v_{id}^{(t)})$  denote the velocity of a particle<sub>*i*</sub> in the searching area at a time-step *t*. Also, let  $P_i = (p_{i1}, p_{i2}, \dots, p_{id})$  and  $P_g = (p_{g1}, p_{g2}, \dots, p_{gd})$  indicate the preferable choice established by the particle itself and by the swarm, respectively. The new location of the particle is updated as follows:

$$V_{id}^{(t+1)} = w \cdot V_{id}^{(t)} + c_1 r_1 (P_{id} - X_{id}^{(t)}) + c_2 r_2 (P_{gd} - X_{id}^{(t)}); \quad i = 1, 2, \dots, N, \quad (2)$$

$$X_{id}^{(t+1)} = X_{id}^{(t)} + V_{id}^{(t+1)}; \quad i = 1, 2, \dots, N, \quad (3)$$

where the particle moves in a multidimensional space,  $c_1$  and  $c_2$  are positive constants,  $r_1$  and  $r_2$  are random numbers in the range  $[0, 1]$ , and  $w$  is the inertia weight.  $V_i(t)$  controls the improvement process and represents both the personal experience of the particle and swapped knowledge from the other surrounded particles. The personal experience of a particle is usually indicated as a cognitive term, and it represents the approaching from the best local position. The swapped knowledge is indicated as a social term in (2), which represents the approaching from the best global position for the swarm.

**2.3. Stock Market.** The stock market is an exchange where securities and shares are traded at a price organized by the request and supply [26]. According to [27], the stock market

is defined by investors' reaction, which is made by the information that is associated with the "real value" of companies. The stock market is one of the most important ways of building wealth. Stocks are the cornerstone of any investment portfolio.

The most significant stock market data called technical data, including the closing price, the high price of a trading day, the low price of a trading day, and the volume of the stock market shares traded per day [28]. A stock index is a measure of a stock market that enables investors to compare between the new and old prices to know the market performance [29]. There are many common indices such as Standard & Poor's 500 (S&P 500), National Association of Securities Dealers Automated Quotations 100 (NASDAQ-100), Dow Jones Industrial Average (DJIA), Gold, and Canadian Dollar to United State Dollar (CADUSD) used with the stock market [30]. The historical data of each of these indices, including the technical data, can be used as a dataset to train the ANN.

### 3. Related Work

A survey of the relevant work shows that there are many studies about forecasting the stock market's future values. The recent technological advances that support AI have led to the appearance of a new trend of investing. In this regard, the buy and sell decisions can be made by treating big data amounts of historical data for stock market indices and determining the risks related to them with high accuracy. Many models fall within this trend such as (1) a dynamic trading rule based on filtered flag pattern recognition for stock market price forecasting [31], (2) stock market trading rule based on pattern recognition and technical analysis [32], (3) intelligent pattern recognition model for supporting investment decisions in stock market [33], (4) using support vector machine with a hybrid feature selection method to the stock trend prediction [34], (5) forecasting stock market trend using machine learning algorithms [35], and (6) deep learning for stock market prediction [36]. For example, the author in [37] was the first to attempt to use ANN in order to model the economic time series of IBM company. This method tries to understand the nonlinear regularity for changes in asset prices like day-to-day variations. However, the scope of the work was limited as the author used a feedforward neural network with only one hidden layer with five hidden units in it without considering the ability to used variant numbers of hidden layers or hidden nodes. Also, the author did not study the effect of the activation function or the learning rate.

In another study [38], the authors proposed a new forecasting method to predict Bitcoin's future price. In this method, PSO was used to select the best values for the number of hidden units, the input lag space, and output lag space of the Nonlinear Autoregressive with Exogeneous Inputs model. The results showed the ability of the model to predict Bitcoin prices accurately. It is worth noting that the authors of [38] did not discuss the impact of the NHL, the ACTFUN, and the learning rate on the performance of the algorithm. However, the proposed technique considered

NHL and used different values for NHL that allows selecting the best hyperparameters of ANN. The proposed model also studied the impact of the activation function and the LR.

A flexible neural tree (FNT) ensemble technique is proposed in [39]. The authors in [39] developed a trusted technique to model the attitude of the seemingly chaotic stock markets. They used a tree structure based evolutionary algorithm and PSO algorithms to optimize the structure and parameters of FNT. Nasdaq-100 index and the S&P CNX NIFTY stock index are used to evaluate their technique. Authors of [39] did not consider the impact of the depth of the neural tree, that is, NHL. They also did not study the impact of other hyperparameters such as ACTFUN, the LR, and the number of children at each level, in other words, the number of hidden nodes.

In [40], the author presented a method based on Multilayer Perceptron and Long Short-Term Memory Networks to predict stock market indices. The historical data of the Bombay Stock Exchange (BSE) Sensex from the Indian Stock market were used to evaluate the method. In [40], the selected NN architecture was verified manually not automatically during a trial-and-error process. The author also used only one hidden layer and changed only the number of hidden units per hidden layer without considering the activation function and the learning rate. However, the proposed technique selected a suitable structure from all possible structures that increase the probability to select the optimal structure and used multiple hidden layers and multiple nodes per hidden layer. The proposed technique changed NHL and NNPHL to adjust selecting the optimal structure and considered the effect of ACTFUN and the LR.

In another work, the authors of [41] suggested an enhanced PSO to train the Sigmoid Diagonal Recurrent Neural Networks (SDRNNs) to optimize the parameters of SDRNN. The historical date of NASDAQ-100 and S&P 500 stock market indices was used to evaluate their model. The authors in [41] did not discuss selection of the best structure of the network in terms of NHL, NNPHL, ACTFUN, and LR.

The authors in [42] presented a new stock market prediction method and applied the firefly algorithm with an evolutionary method. They implemented their method to the Online Sequential Extreme Learning Machine (OSELM). Their model's performance was evaluated using the datasets of BSE Sensex, NSE Sensex, S&P 500, and FTSE indices. The author of [42] did not optimize the hyperparameters of the network.

In [43], the authors suggested a new end-to-end hybrid neural network approach to forecast the stock market index's price direction. This approach depends on learning multiple time scale features extracted from the daily price and CNN network to perform the prediction using completely linked layers that bring together features learned by the Long Short-Term Memory network. The historical data of the S&P 500 index was used to evaluate this approach. The author of [43] did not study the effect of optimization of the hyperparameters on the network's performance.

The authors of [44] proposed a hybrid model to forecast the Nikkei 225 index price for the Tokyo stock market. The hybrid model used ANN and genetic algorithms to optimize

the accuracy of stock price prediction. ANN was trained using a backpropagation as a learning algorithm. In [44], the authors used ANN with only one hidden layer and did not consider the effect of hyperparameters of ANN such as NHL, NNPHL, ACTFUN, and LR, while in [45], the authors proposed an adaptive system to predict the price in the stock market. This system uses the PSO algorithm to overcome the problems of the backpropagation approach and to training ANN to forecast the price of the S&P 500 Index and the NASDAQ Composite Index, therefore helping the investors to make correct trading decisions. They also used ANN with only one hidden layer with a fixed NNPHL equal to  $2N-1$ , where  $N$  is the number of inputs. They did not consider the effect of other hyperparameters such as ACTFUN and LR.

In [46], the authors studied the different modifications on PSO and its application over the stock market to predict the prices. In another work [47], the PSO algorithm selects the optimally weighted signals set of trading signals that specify buy order, sell order, or hold order depending on evolutionary learning from the New York Stock Exchange (NYSE) and the Stock Exchange of Thailand (SET).

In [48], the authors proposed an effective prediction model using PSO to forecast the S&P 500 and DJIA Stock Indices for the short and long term. This model applies an adaptive linear combiner (ALC), whereas PSO modifies its weights. The authors compared their model with the MLP-based model. The results show that their model better than the MLP-based model in terms of accuracy and training time.

#### 4. The Proposed Technique

The neural network parameters' determination to create an appropriate architecture of the neural network that produces output with accepted error is very time- and cost-consuming. To determine the architecture of the network, the configuration is generally carried out by hand, in a trial-and-error fashion. However, the manual tuning of the hyperparameters of ANN through the trial-and-error process and finding accurate configurations consume a long time. A different approach uses some global optimization techniques, for instance, applying the PSO algorithm to choose the best architectures with minimum error. As can be seen from the discussion of the related literature, most of the proposed methods suffer from some limitations, such as using certain fixed NHL, a fixed NNPHL, or the number of iterations. In most of these methods, the proposed approaches study the effect of only one of these hyperparameters simultaneously. Some of the suggested algorithms are time-consuming and have a high computational cost. The particle swarm optimization was very usefully employed in finding optimal parameters [49]. For that, in our earlier work [50], the effect of NHL and NNPHL on the performance of ANN using PSO was studied and discussed.

The purpose of the proposed model is to provide a potential solution for the problem of designing the best structure for multilayer ANN (MLANN) via selecting the best configuration for the network, for example, choosing the best hyperparameters, including NHL, NNPHL,

ACTFUN, and the learning rate, maximizing prediction accuracy and minimizing processing time.

Now, we modify the standard PSO by adding the ‘‘center of gravity (CoG)’’ concept which can be characterized as a mean of gravities calculated by their distances from a referenced center. A new algorithm called PSO with center of gravity (PSOCoG) is proposed in this paper as described in the following paragraphs. Assume that there are  $N_g$  single gravitational points  $1, \dots, N_g$ . The movement of these gravitational individuals is identified by appointing their location vectors  $\vec{s}_1, \dots, \vec{s}_{N_g}$  as follows:  $\vec{s}_i(t)$ , where  $i = 1, \dots, N_g$ . CoG is a position in a gravities system where the location vector  $\vec{S}$  is determined utilizing the gravities  $\mathbf{g}_i$  and location vectors  $\vec{s}_i$  in this way:

$$\vec{S} = \frac{1}{G} \sum_{i=1}^{N_g} g_i \times \vec{s}_i, \quad (4)$$

$$G = \sum_{i=1}^{N_g} g_i, \quad (5)$$

where  $G$  is the sum of gravities and  $N_g$  is the number of gravities [51].

In this technique, an efficient CoG-particle ( $X_{cg}$ ) is suggested.  $X_{cg}$  will participate in speeding up the convergence of the model in fewer repetitions. It helps in finding a closer solution to the optimal and enhancing the quality of the solution. The CoG-particle is a virtual member in the swarm used to express the swarm at each repetition. In addition, CoG-particle is weighted via the values of a fitness function for the swarm members. It has no speed and does not involve any of the standard swarm member’s tasks, like fitness valuation, searching for the optimal solution. The weighted center of the swarm can be determined as follows through considering the swarm individuals as a group of gravity points and making the value of the target function of each individual match the gravity:

$$X_{cg}(t) = \frac{1}{F_{cg}} \sum_{i=1}^{N_g} F(x_i) \times x_i(t), \quad (6)$$

where  $F(x_i)$  is the fitness value of member  $i$  at location  $x_i(t)$  and  $F_{cg}$  is the sum of the values of fitness for all members. Using formula (7),  $F_{cg}$  can be determined similar to the sum of gravities  $G$  in formula (5). The fitness value  $F(x_i)$  for maximum optimization has been given in formula (8), while formula (9) is used for minimum optimization:

$$F_{cg} = \sum_{i=1}^{N_g} F(x_i), \quad (7)$$

$$F(x_i) = f(x_i), \quad (8)$$

$$F(x_i) = \frac{1}{f(x_i) + \varepsilon}, \quad \varepsilon \cong 0, \quad (9)$$

where  $f(x_i)$  is the objective function and its range supposed to be positive in this situation.

The updated speed formula for each member can be created via equation (10) by computing  $X_{cg}$ , the best global solution at the whole swarm level  $P_{gs}$ , and the best local solution detected by every individual  $P_{il}$  as follows:

$$v_{id}^{(t+1)} = wv_{id}^{(t)} + c_1r_1 \left( \frac{P_{ild}^{(t)} + X_{cg}^{(t)}}{2} - X_{id}^{(t)} \right) + c_2r_2 \left( \frac{P_{gsd}^{(t)} - X_{cg}^{(t)}}{2} - X_{id}^{(t)} \right), \quad i = 1, N_g. \quad (10)$$

Then, the modified location for any individual in the swarm is calculated using

$$X_{id}^{(t+1)} = v_{id}^{(t+1)} + X_{id}^{(t)}, \quad i = 1, N_g, \quad (11)$$

where the particle moves in a multidimensional space. The searching cycle proceeds till the termination condition is true. The idea behind that is to make such a CoG-particle ( $X_{cg}$ ) the convergence of swarm individuals towards the optimal global solution and help them move away from the local solution. It may be possibly clarified by explaining the function of the second and third parts in (10). The term  $(P_{ild}^{(t)} + X_{cg}^{(t)}/2 - X_{id}^{(t)})$  is guidable for the attractiveness of the present location of the individual in direction of the average of the positive direction of its best local solution ( $+P_{ild}$ ) and the positive direction of the location of CoG-particle ( $+X_{cg}$ ), which assists the personal experience term to move away from the local solutions, while the term  $(P_{gsd}^{(t)} - X_{cg}^{(t)}/2 - X_{id}^{(t)})$  is guidable for the attractiveness of the present location of the swarm member in the direction of the average positive direction towards the best global solution ( $+P_{gs}$ ) and the negative direction of the location of CoG-particle ( $-X_{cg}$ ), which assists keeping the efficiency of the population diversity (exploration and exploitation process) through the searching operation. This increments the probability of rapidly approaching global solutions (or near-global solutions), where the CoG-particle will pull in swarm members to the area of best-discovered promising solutions. Consequently, that offers individuals the optimal opportunity to take the location of the global best-discovered candidate solution through the discovery phase. All earlier motions are propped by linearly decreasing weight, which allows the ability to adjust the population diversity through the searching operation.

The proposed prediction model used a new PSOCoG technique to train ANN and the proposed model called the ANN-PSOCoG model. In the ANN-PSOCoG model, a 4-dimensions search space was used. The first dimension is the number of HL, the second dimension is NPHL, the third dimension is the type of activation function (ACTFUN), and the fourth dimension is the learning rate. Any particle inside the discovery area is considered as a candidate solution. This means that the location of the particle determines the values of NHL, NNPHL, ACTFUN, and LR, which represent a possible configuration for the network. In the search phase, the PSO technique is used to find the best settings by flying the particles within a bounded search space. Each particle

```

(1) Start
(2) Setting up the swarm members randomly;
(3) Run NN corresponding to each particle;
(4) while (termination condition is not met)
(5)   for  $i = 1$  to pop
(6)     Compute the fitness value ( $f_i$ ) for each particle;
(7)     if (the fitness value is less than  $P_{il}$ )
(8)       Set current fitness value as the new ( $P_{il}$ );
(9)     endif
(10)    select the minimum ( $f_i$ ) of all particles in swarm as  $P_{gs}$  ;
(11)    for  $d = 1$  to D
(12)      Calculate  $V_{id}^{(t+1)}$  from Equation (10);
(13)      Calculate  $X_{id}^{(t+1)}$  from Equation (11);
(14)    end-for
(15)  end-for
(16)  return the best particle with gbest;
(17) end-while
(18) Run the NN corresponding to the best selected hyperparameters; 19: end-algorithm.

```

ALGORITHM 1: The proposed model.

has its attributes, which are location, speed, and fitness value calculated by a fitness function. The particle's speed defines the next movement (direction and traveled distance). The fitness value represents an index for the convergence of the particle from the solution. The position of each particle is updated to approach towards the individual that has an optimal location according to (10) and (11). In every repetition, every particle in the swarm modifies its speed and location based on two terms: the first is the individual optimal solution, which is the solution that the particle can get personal. The second is the global optimal solution that is the solution that the swarm can obtain cooperatively till now. The suggested technique used several particles (pop) which are initialized randomly. For each particle, the corresponding ANN is created and evaluated using a fitness function shown in

$$f_i = \frac{1}{m} \sum_1^m (\text{expected output}_i - \text{actual output}_i)^2, \quad (12)$$

where  $m$  is the number of inputs for ANN.

The fitness values of all swarm members are determined using (12). For each particle, the location was stored as  $X_i$  and the fitness value was stored as  $P_{il}$ . Among all  $P_{il}$ ,  $P_{il}$  with minimum fitness value is selected as the global best particle ( $P_{gs}$ ); then the location and fitness value of  $P_{gs}$  are stored. This process is repeated by updating the particle positions and velocities according to (10) and (11). The process is iterated till the best solution is found or the maximum number of iterations (maxite) is reached. The global best particle represents the best selected hyperparameters that are used to build ANN. Algorithm1 explains the suggested model.

## 5. Results and Performance Evaluation

In this section, the evaluation of the performance of the new model is presented. The dataset and settings of the

experiment are described in detail and the results of the proposed model are discussed.

*5.1. Experimental Datasets.* The historical data of Standard's and Poor's 500 (S&P 500) [52] and Dow Jones Industrial Average (DJIA) [53] were used as the datasets for the stock market prediction experiments to evaluate the ANN-PSOCOG model.

The 500 stocks that highlight the performance of large-cap entities, selected by leading economists and weighted by market values, are referred to as the Standard and Poor's 500 Index. The S&P 500 Index or the Standard & Poor's 500 Index is a market-capitalization-weighted index of 500 of the largest publicly traded companies in the US stock market. The S&P 500 is a leading indicator and one of the most prevalent benchmarks and regarded as the best gauge of large-cap US equities [54].

The Dow Jones Industrial Average (DJIA) is a stock market index that measures the stock performance of 30 large companies listed on stock exchanges in the United States such as Apple Inc., Cisco Systems, The Coca-Cola Company, IBM, Intel, and Nike. It is considered one of the most commonly followed equity indices in the USA. Two-thirds of the DJIA's companies are manufacturers of industrial and consumer goods while the others represent diverse industries. Besides longevity, two other factors play a role in its widespread popularity. It is understandable to most people, and it reliably indicates the market's basic trend [55].

To evaluate the proposed model, we used the historical data of S&P 500 dataset, where the overall number of observations for the exchange indices was 3776 trading days, from August 16, 2005, to August 16, 2020. Also, we used historical data of DJIA dataset where the overall number of observations for the exchange indices was 9036 trading days, from Jan 28, 1985, to Dec 02, 2020. Each observation includes the opening price, highest price, lowest price, the

closing price, and the total volume of the stocks traded per day. The opening price, highest price, lowest price, and the total volume of the stocks traded per day were used as inputs of the artificial neural network, while the closing price was used as output for ANN. In all scenarios discussed in this paper, the data are divided as 80% of the available data that were used as a training set, while 10% of the available data were used as a validation set, and the last 10% of the available data were used as a test set.

**5.2. Parameter Settings.** To run the proposed method, PSOCoG and multilayer neural network settings were set as the swarm size (pop)=20 and the maximum number of iterations (maxite)=100. Besides, the following settings are used:

- (i) The values of  $c_1$  and  $c_2$  were set to 2 [56]
- (ii) The value of  $W_{min}$  was set to 0.4 [56]
- (iii) The value of  $W_{max}$  was set to 0.9 [56]
- (iv) The inertia weight ( $W$ ) was linearly decreased from  $W_{max}$  to  $W_{min}$  [57]
- (v) 4-dimensional search space (HL and NPHL, ACTFUN, and LR) is used to represent the hyperparameter of ANN
- (vi) Initialization ranges for HL were set to [1 7] according to [44, 58]
- (vii) NPHL was set to [14 21] according to [59, 60]
- (viii) ACTFUN was set to {'logsig','softmax','tansig','purelin'}
- (ix) LR was set to and [0.01 0.9] [61]
- (x) Levenberg–Marquardt algorithm [62] was used as the train function for the network

In this study, Mean absolute percentage error (MAPE), mean absolute error (MAE), relative error (RE), and mean-square error (MSE) shown in (13) to (16) [63] are used to evaluate the accuracy of the proposed model:

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{\text{actual output}_i - \text{expected output}_i}{\text{actual output}_i} \right| \times 100\%, \quad (13)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |\text{actual output}_i - \text{expected output}_i|, \quad (14)$$

$$RE_i = \frac{\text{actual output}_i - \text{expected output}_i}{\text{actual output}_i}, \quad (15)$$

$$MSE = \frac{1}{n} \sum_{i=1}^n (\text{actual output}_i - \text{expected output}_i)^2. \quad (16)$$

**5.3. Results and Discussion.** To evaluate the performance of the proposed model and verify its effectiveness, several simulations were conducted in four different scenarios:

- (1) In the first scenario, ANN was used
- (2) In the second scenario, the standard PSO (SPSO) was used
- (3) In the third scenario, the assemble of SPSO with CoG (SPSOCoG) was used
- (4) In the fourth scenario, the assemble of ANN with SPOS was used
- (5) In the fourth scenario, the proposed model ANN-PSOCoG was used

ANN-PSOCoG also is compared with other models, that is, ANN, SPSO, and ANN-SPSO. Finally, the impact of COVID-19 on the proposed model was studied.

The experiments were executed using a PC with Windows 10 operating system and Intel Core i5 processor running at 3.30 GHz and 8 GB of RAM.

**5.3.1. Scenario 1: ANN Experiments.** In this scenario, only the artificial neural networks were used to predict the closing price. The historical data of Standard's and Poor's 500 (S&P 500) were used as the dataset to train the ANN. Since the Levenberg–Marquardt function has a fast convergence rate, it was used to train the proposed artificial neural network.

To evaluate this model, MAPE, MAE, and RE were calculated. The computed MAPE for this model is 0.007%, while MAE is 0.087. Figure 2(a) shows the actual closing price (ACP) and the expected closing price for this model, while Figure 2(b) shows the relative error (RE) for the ANN model.

**5.3.2. Scenario 2: Standard PSO (SPSO) Experiments.** In this scenario, only standard particle swarm optimization (SPSO) was used to predict the closing price using historical data of S&P 500. The size of the swarm was 20, and the maximum number of iterations was 100. The values of the other parameters were set as mentioned above. MAPE for this method is 0.0053%, while MAE for this method is 0.070%. The comparison between the actual closing price and expected closing price using SPSO (ECP-SPSO) is shown in Figure 3(a), while the relative error for this method is shown in Figure 3(b).

**5.3.3. Scenario 3: SPSOCoG Experiments.** In this scenario, the standard particle swarm optimization (SPSO) with CoG concept was used to predict the closing price using historical data of S&P 500. The size of the swarm was 20, and the maximum number of iterations was 100. The values of the other parameters were set as mentioned above. MAPE for this method is 0.0038%, while MAE for this method is 0.067%. The comparison between the actual closing price and expected closing price using SPSOCoG (ECP-SPSOCoG) is shown in Figure 4(a), while the relative error for this method is shown in Figure 4(b).

**5.3.4. Scenario 4: ANN-SPSO Experiments.** In this scenario, a combination of the artificial neural network and the standard particle swarm optimization (ANN-SPSO) was

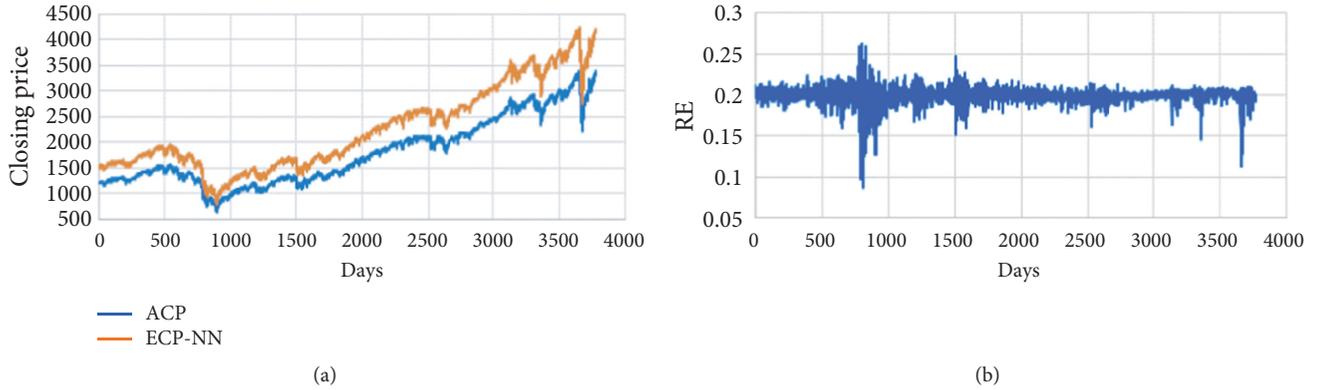


FIGURE 2: ANN model performance. (a) ACP versus ECP-ANN. (b) The relative error for ANN model.

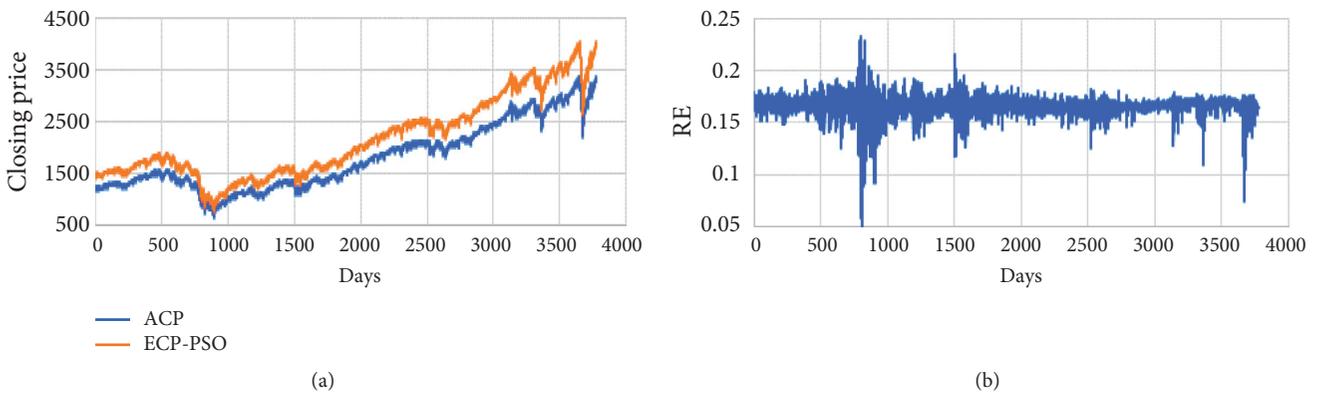


FIGURE 3: SPSO model performance. (a) ACP versus ECP-SPSO. (b) The relative error for SPSO.

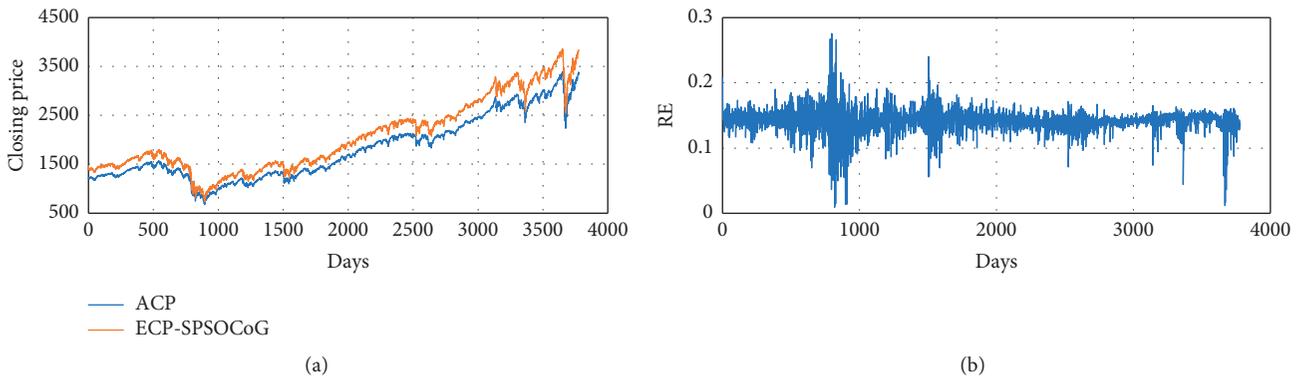


FIGURE 4: SPSOCoG model performance. (a) ACP versus ECP-SPSOCoG. (b) The relative error for SPSOCoG.

used to predict the closing price, where SPSO was used to train the network to select the best configurations. Dataset of historical data of S&P 500 was used in this scenario. MAPE for ANN-SPSO is turned to be 0.0027%, while MAE for ANN-SPSO equals 0.047. Figure 5(a) depicts ACP and expected closing price for ANN-SPSO (ECP-ANN-SPSO), while RE for this model is shown in Figure 5(b).

5.3.5. *Scenario 5: ANN-PSOCoG Experiments.* In this scenario, to design the best architecture of the ANN, the proposed model (ANN-PSOCoG) was running with the chosen settings to select the best hyperparameters for the network in terms of the HL, NPHL, ACTFUN, and LR. Dataset of historical data of S&P 500 was used to train the network. The result of running the ANN-PSOCoG model displays that the best configuration was chosen by particle 7

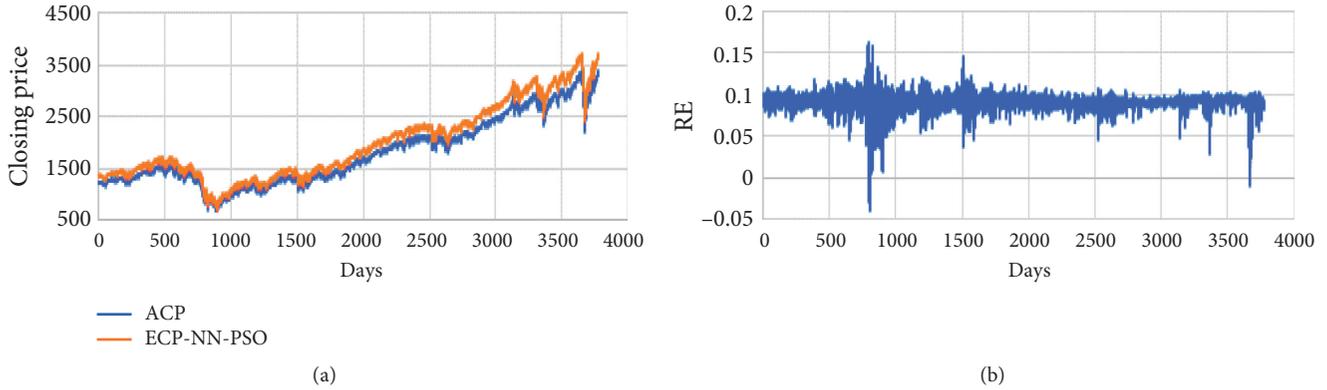


FIGURE 5: ANN-SPSO model performance. (a) ACP versus ECP-ANN-SPSO. (b) RE for ANN-SPSO model.

with MPAE equal to 0.00024% and MAE equal to 0.00017. The prediction accuracy of ANN-PSOCoG is very high, as seen from MPAE and MAE values, in which they are very small and close to zero, the corresponding hyperparameters, i.e., NHL was equal to 6, the NNPHL was equal to 21, the selected activation function was purelin, and the learning rate was equal to 0.5469.

Although the process of determining the best hyperparameters of the ANN consumes a long time [64], the elapsed time to find the best configuration in the ANN-PSOCoG model was only 51.082822 seconds. ANN-PSOCoG can be considered, in light of this, efficient in terms of processing time.

The value of regression ( $R$ ) shows the correlation between the forecasted outputs and targets. The regression plot of ANN-PSOCoG is shown in Figure 6. The value of the regression coefficient ( $R$ ) for training was 0.99995, validation was 0.99992, and test was 0.99994, while for all was 0.99994. In this case, the prediction accuracy of ANN-PSOCoG can be considered very high since the forecasted output via the proposed model ANN-PSOCoG is almost equal to the target. Figure 7(a) displays the actual closing price and expected closing price using ANN-PSOCoG (ECP-ANN-PSOCoG). The relative error of the proposed model is shown in Figure 7(b). As can be seen in the figure, the proposed model predicts the closing price with a very high accuracy that is close to the actual closing price. Based on the above discussion, it is noted that the proposed model is able to give investors the trust to make the correct decisions to buy or sell shares that achieve the largest possible profit and avoid risk and loss. The mean absolute percentage error (MAPE) represents the risk in the stock market and MAPE represents the fluctuation of ECP around ACP. So, if the fluctuation of the price increases, the error of prediction increases, and the risk of the prediction of the closing price for the stock market increases.

Besides, MAPE for ANN-PSOCoG was very small, so the risk in the proposed model is very small and prediction accuracy is very high.

**5.3.6. Comparison of ANN-PSOCoG with ANN, SPSP, SPSoCoG, and ANN-SPSO.** To study the efficiency of the ANN-PSOCoG model, a comparison of this model with the

ANN-SPSO, SPSoCoG, SPSo, and ANN was carried out using historical data of S&P 500 and DJIA datasets. Figure 8 shows the comparison between all models' expected closing price and the actual closing price using the historical data of S&P 500. As can be seen in the figure, ANN-PSOCoG outperformed ANN-SPSO in terms of prediction accuracy by approximately 13%, while it outperformed SPSoCoG by about 17%; also it outperformed SPSo by about 20% and ANN-PSOCoG outperformed ANN by approximately 25%.

Figure 9(a) shows the comparison between all models' expected closing price and the actual closing price using the historical data of DJIA dataset. As can be seen in the figure, ANN-PSOCoG outperformed ANN-SPSO in terms of prediction accuracy by approximately 18%, while it outperformed SPSoCoG by about 24%; also it outperformed SPSo by about 33% and ANN-PSOCoG outperformed ANN by approximately 42%, while Figure 9(b) shows the relative error (RE) for all models.

Tables 1 and 2 show the elapsed time (ET) which is the time to find the best configuration, MAPE and MAE for ANN-PSOCoG, ANN-SPSO, SPSPCoG, SPSo, and ANN models using the historical data of S&P 500 and DJIA datasets, respectively.

As shown in Tables 1 and 2, the proposed ANN-PSOCoG performance is the best with the lowest values of MAPE and MAE compared to other models for both the S&P 500 and DJIA datasets. By contrast, the ANN model displays the highest values of MAPE and MAE. Therefore, the proposed model's accuracy is the best against other models, and ANN-PSOCoG is more efficient than other models in terms of ET for both the S&P 500 and DJIA datasets.

To show the quality of the considering methods and their convergence to the best solution, MSE is used as a metric through the training phase. Figure 10(a) shows MSE for the compared models using S&P 500 dataset, while Figure 10(b) shows MSE for the compared models using DJIA dataset.

It is noted from Figure 10 that the suggested ANN-PSOCoG converged more rapidly than the remaining models and achieved the best minimum value of MSE for both the S&P 500 and DJIA datasets. This means that the prediction quality of the proposed model is the highest compared to other models.

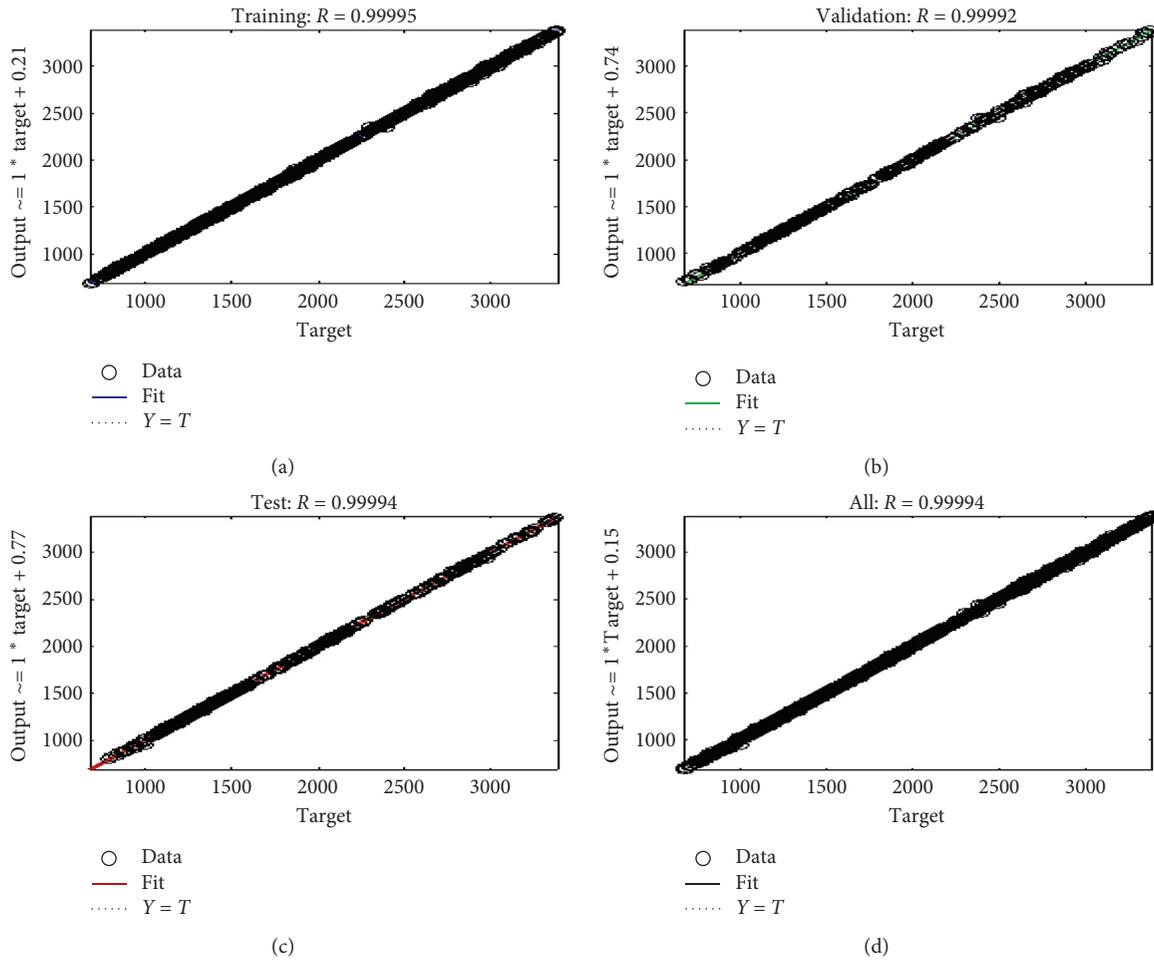


FIGURE 6: The regression values for ANN-PSOCoG.

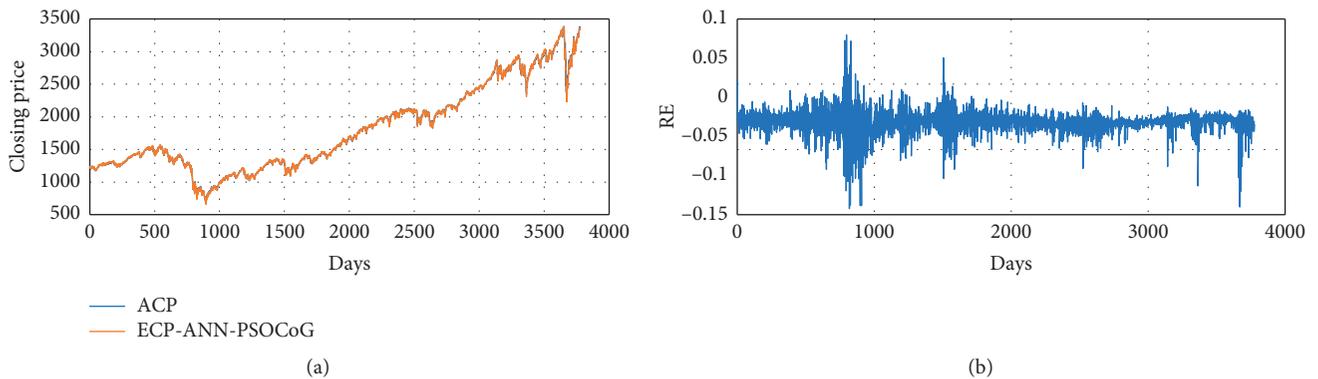


FIGURE 7: ANN-PSOCoG model performance. (a) ACP versus ECP-ANN-PSOCoG. (b) RE for ANN-PSOCoG.

5.3.7. *Study of Efficient ANN-PSOCoG under COVID-19.* The rapid spread of coronavirus (COVID-19) has severe impact on the global stock markets. It has created an unmatched level of risk, resulting in investors suffering considerable losses in a very short time. For example, the drop of the S&P 500 index in the USA during the COVID-19 crisis, where it lost one-third of its value during only one month [62]. Figure 11 shows the comparison between the drop of the S&P 500 index during the dotcom crisis (which peaked

on March 24, 2000), the subprime crisis (peaked on Oct. 9, 2007), and the COVID-19 crisis (peaked on Feb 19, 2020) [65]. As it can be seen from the figure, in March 2020, it took only one month for the S&P 500 to lose one-third of its value, while it took one year for the subprime crisis to decline the same amount and one year and a half for the dotcom bust. Till September 2, 2020, there have been about 25.6 million positive cases of coronavirus globally including 852,758 deaths. The pandemic has affected more than 210

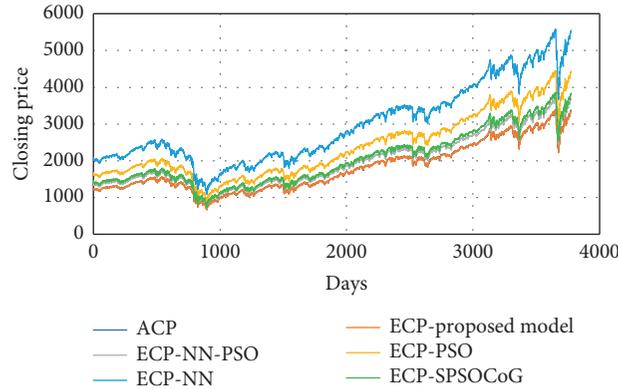


FIGURE 8: The comparison between the proposed model and other models.

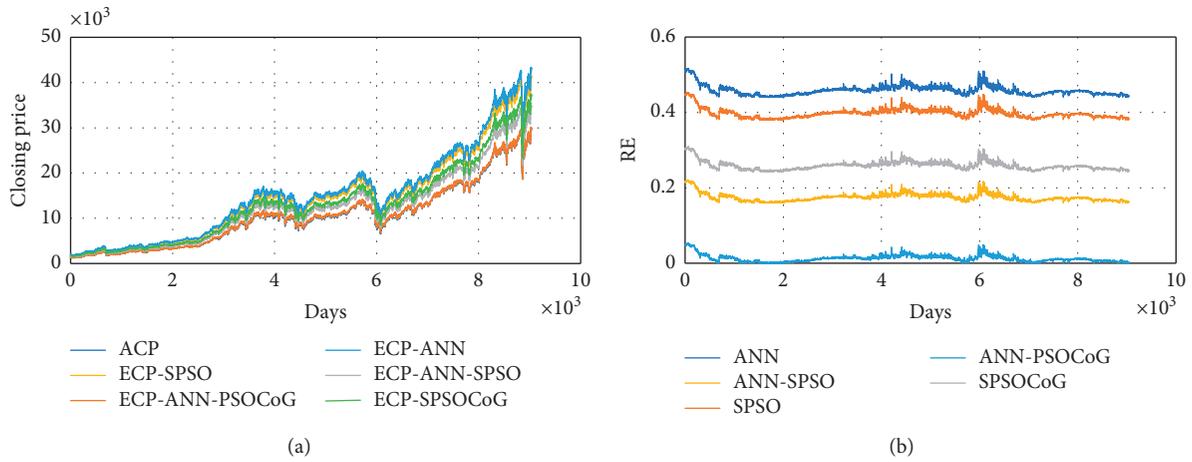


FIGURE 9: Performance of all models using DJIA dataset. (a) The comparison between the proposed model and other models. (b) RE for all models using DJIA dataset.

TABLE 1: MAPE, MAE, and ET for ANN\_PSOCoG, ANN-SPSO, SPSOCOg, SPSO, and ANN models using S&P 500 dataset.

Metrics	ANN-PSOCoG	ANN-SPSO	SPSO COG	SPSO	ANN
MAPE%	0.00024	0.0027	0.0038	0.0053	0.0070
MAE	0.00017	0.047	0.067	0.070	0.087
Elapsed time (sec)	51.082822	89.128091	95.21345	112.375813	135.987016

TABLE 2: MAPE, MAE, and ET for ANN\_PSOCoG, ANN-SPSO, SPSOCOg, SPSO, and ANN models using DJIA dataset.

Metrics	ANN-PSOCoG	ANN-SPSO	SPSO COG	SPSO	ANN
MAPE%	0.00072	0.030	0.042	0.062	0.071
MAE	0.00055	0.0024	0.0033	0.0049	0.0056
Elapsed time (sec)	83.4259	97.3215	105.6284	138.9368	153.7569

countries [66]. Based on the above, finding an accurate prediction model for the stock market is very important. For that, the proposed ANN-PSOCoG model is presented and tested using historical data from December 31, 2019, to August 14, 2020, including the period of spread the COVID-19, of different stock market indices such as S&P 500, Gold, NASDAQ-100, and CANUSD [67].

Figure 12(a) displays the result of applying the proposed ANN-PSOCoG over historical data of S&P 500. Figure 12(b)

shows the relative error for ACP and ECP for ANN-PSOCoG over historical data of S&P 500.

Figure 13(a) shows ACP and ECP for ANN-PSOCoG over historical data of Gold, while Figure 13(b) shows RE for ANN-PSOCoG over historical data of Gold.

ACP and ECP for tested ANN-PSOCoG using historical data of NASDAQ-100 are shown in Figure 14(a), while RE of ANN-PSOCoG using data of NASDAQ-100 is shown in Figure 14(b).

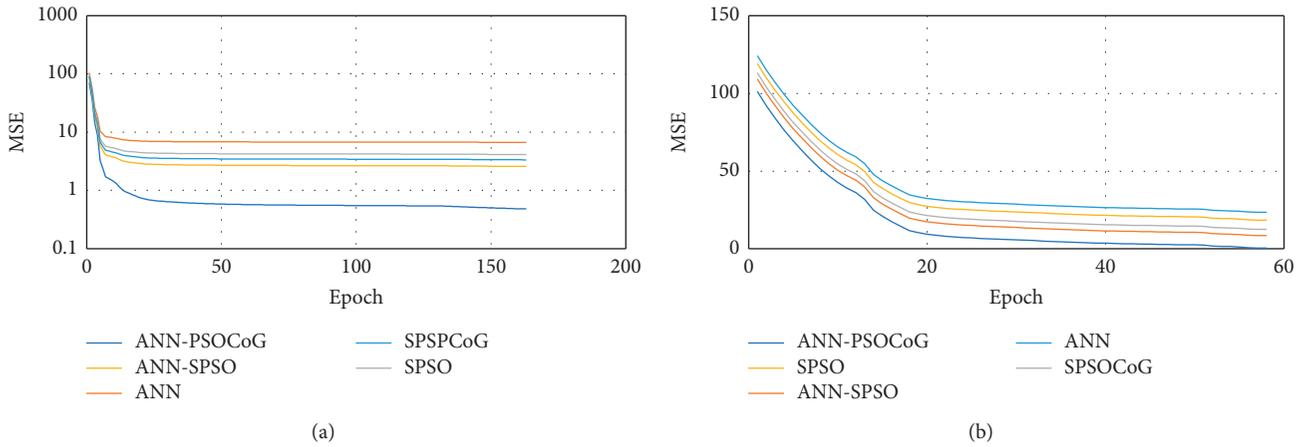


FIGURE 10: MSE for all models. (a) MSE for all models using S&P 500 dataset. (b) MSE for all models using DJIA dataset.

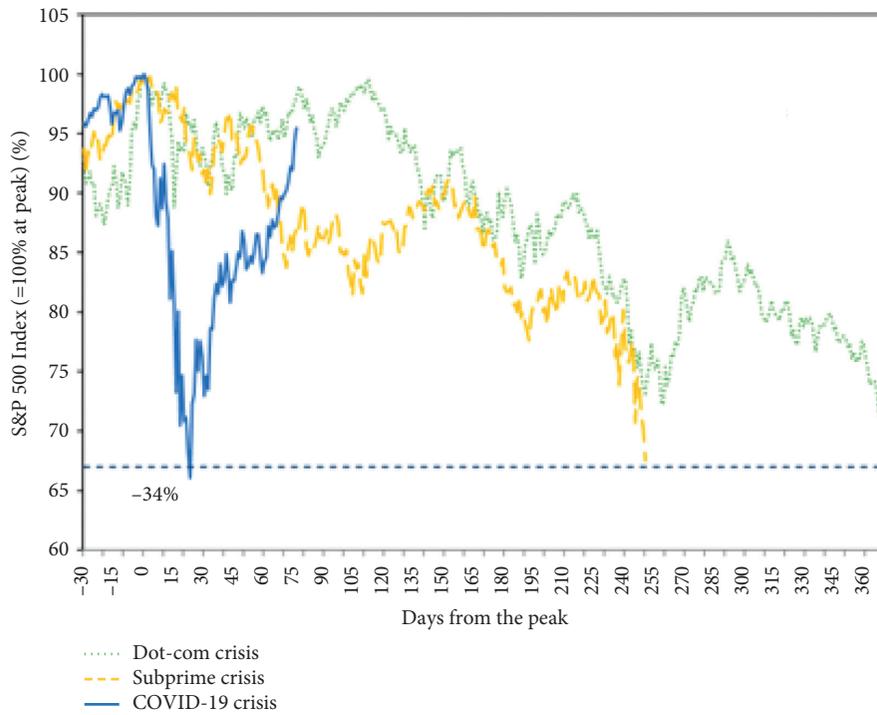


FIGURE 11: Effect of the COVID-19 on stock market in USA.

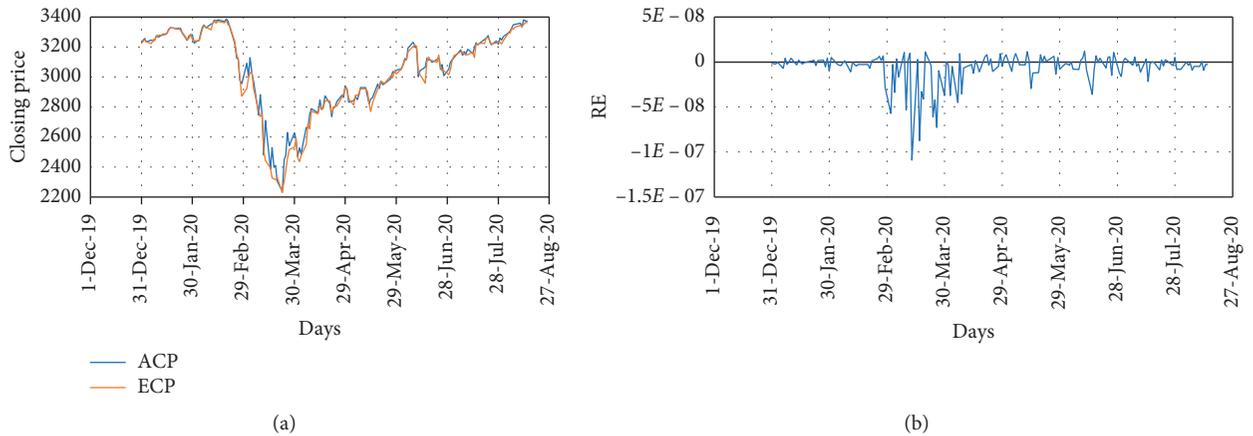


FIGURE 12: Performance of ANN-PSOCoG over historical data of S&P 500. (a) ACP versus ECP using ANN-PSOCoG. (b) RE for ANN-PSOCoG.

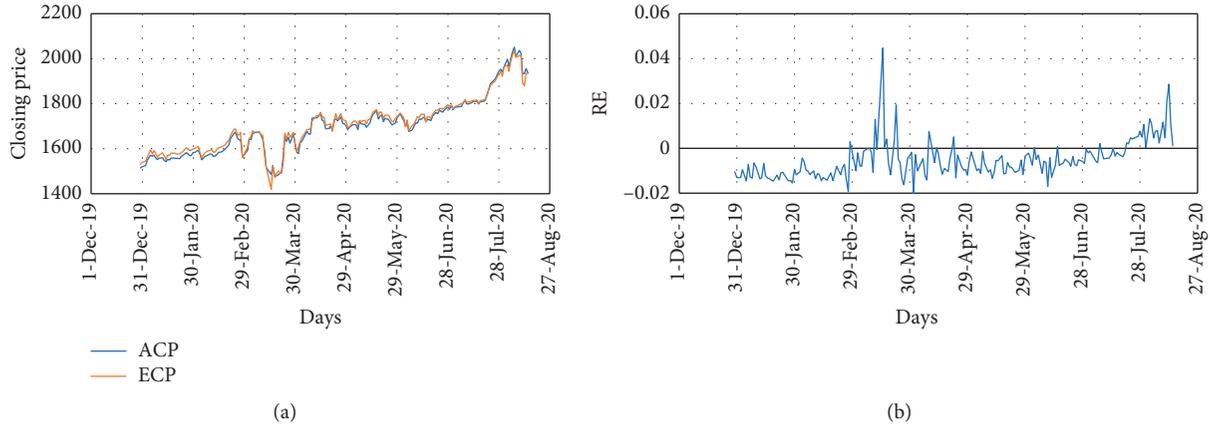


FIGURE 13: Performance of ANN-PSOCoG over historical data of Gold. (a) ACP versus ECP using ANN-PSOCoG. (b) RE for ANN-PSOCoG.

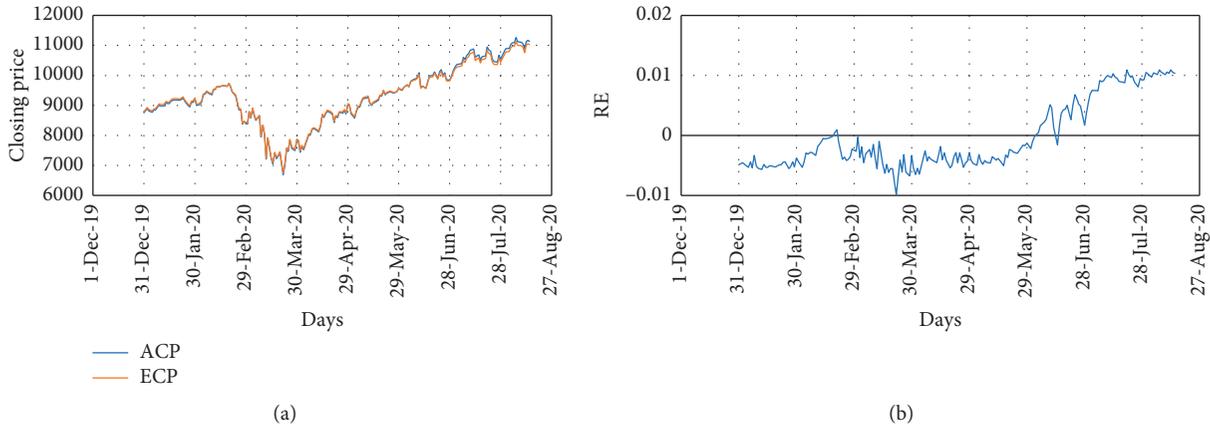


FIGURE 14: Performance of ANN-PSOCoG over historical data of NASDAQ-100. (a) ACP versus ECP using ANN-PSOCoG. (b) RE for ANN-PSOCoG.

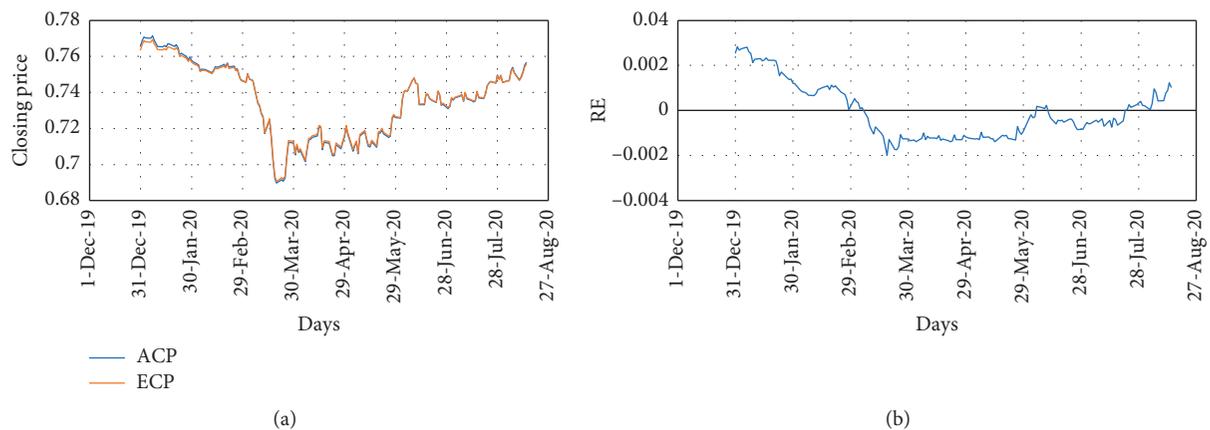


FIGURE 15: Performance of over historical data of CANUSD. (a) ACP versus ECP using ANN-PSOCoG. (b) RE for ANN-PSOCoG.

The last test was training the proposed model using the historical data of CANUSD which represents the ratio between the Canadian dollar and the American dollar. The actual rate and expected rate are shown in Figure 15(a), while the prediction's relative error is shown in Figure 15(b).

MAE and MAPE for the proposed ANN-PSOCoG using the historical data of S&P 500, Gold, NASDAQ-100, and CANUSD are shown in Table 3.

As seen from Table 3 and from Figures 11–14, the results of the proposed ANN-PSOCoG showed a high accuracy of

TABLE 3: MAPE and MAE for S&amp;P 500, Gold, NASDAQ-100, and CANUSD datasets.

Metrics	S&P 500	Gold	NASDAQ-100	CANUSD
MAPE %	0.00158	0.007799	0.00239	0.001719891
MAE	0.030529	0.079009	0.224702	1.19491E-05

prediction for the closing price using different historical data of stock market indices during the peak spread of COVID-19. These results assure the ability of the proposed ANN-PSO-CoG to predict the closing price with very high accuracy and confirm the efficiency of the proposed model under the effect of coronavirus pandemic.

## 6. Conclusion and Future Work

In this paper, a new modification of particle swarm optimization using the “center of gravity (CoG)” concept (PSOCoG) is proposed. This modification gives a new efficient search technique. It benefits from the physical principle “center of gravity” to move the particles to the new best-predicted position. The newly proposed technique is used as a decision-making model integrated with ANN as a learning model to form a hybrid model called ANN-PSO-CoG. The decision-making model will help the ANN to select the best hyperparameters values. To evaluate the effectiveness of the ANN-PSOCoG model, the proposed hybrid model was tested using historical data of two different datasets; they are S&P 500 and DJIA. The results show that the proposed model was able to select the best hyperparameters used to construct the desired network with very high accuracy of prediction and with a very small error. The proposed model displayed better performance compared with ANN, SPSO, and ANN-SPSO models in terms of the prediction accuracy. Using S&P 500 dataset, the results show that the ANN-PSOCoG model outperforms ANN model by approximately 25%, SPSO model by approximately 20%, SPSOCoG model by approximately 17%, and ANN-SPSO model by approximately 13%. Also, the results show that the ANN-PSOCoG model outperforms ANN model by approximately 42%, SPSO model by approximately 33%, SPSOCoG model by approximately 24%, and ANN-SPSO model by approximately 18% using DJIA dataset. In terms of elapsed time, the results showed that the proposed model is approximately 1.7 times faster than the ANN-SPSO model, approximately 1.8 times faster than the SPSOCoG model, approximately 2.2 times faster than the SPSO model, and approximately 2.6 times faster than the ANN model using S&P 500 dataset. While using the DJIA dataset, the results showed that the proposed model is approximately 1.2 times faster than the ANN-SPSO model, approximately 1.3 times faster than the SPSOCoG model, approximately 1.7 times faster than the SPSO model, and approximately 1.9 times faster than the ANN model. The proposed ANN-PSOCoG shows a high efficiency under the effect of coronavirus disease (COVID-19). The proposed model was trained using different datasets of stock markets and the results proved the efficiency and accuracy of the proposed ANN-PSOCoG model. The values of MAPE, MAE, and RE were very small for S&P 500, GOLD, NASDAQ-100, and CANUSD datasets.

The proposed model might need more training time and it requires modern computational resources when it is used with a very huge dataset, oil reservoir dataset for instance. Also, when the particle dimensions increase, the complexity will increase.

As future work, some points can be discussed to improve the proposed model as follows:

- (i) We study the effect of changing NNPHL for each layer instead of being the same in all hidden layers in the proposed model
- (ii) We extend the proposed ANN-PSOCoG to discuss the ability to select more hyperparameters such as the number of iterations and the size of a batch
- (iii) We discuss the effect of the activation function if we apply the same activation function for all the layers or apply different activation functions for each layer
- (iv) We study the impact of using more than one swarm for PSOCoG can be considered
- (v) A comparison of the proposed model with other techniques such as ANN-GA or ANN-ACO can be carried out
- (vi) The proposed algorithm (PSOCoG) in combination with Elman neural network (ENN-PSOCoG) could be used to predict the closing price and compare it with ANN-PSOCoG using a huge dataset
- (vii) We discuss more metrics such as root-mean-square error (RMSE) and root-mean-square deviation (RMSD)

## Data Availability

The data used in this study are available on Yahoo Finance website and can be accessed using the following link: <https://finance.yahoo.com>.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Authors' Contributions

Razan Jamous, Mohamed El-Darieby, and Hosam ALRahhal carried out conceptualization and methodology; Razan Jamous and Hosam ALRahhal were responsible for software, formal analysis, data curation, writing, and original draft preparation; Mohamed El-Darieby performed reviewing and editing and supervised and was responsible for funding acquisition. All authors have read and agreed to the published version of the manuscript.

## References

- [1] World Health Organization, “WHO characterizes COVID-19 as a pandemic.” 11 March, 2020b, [https://www.who.int/docs/default-source/coronaviruse/transcripts/who-audio-emergencies-coronavirus-press-conference-full-and-final-11mar-2020.pdf?sfvrsn=cb432bb3\\_2](https://www.who.int/docs/default-source/coronaviruse/transcripts/who-audio-emergencies-coronavirus-press-conference-full-and-final-11mar-2020.pdf?sfvrsn=cb432bb3_2).
- [2] K. Andersen, “Morningstar’s view: the impact of coronavirus on the economy.” stock strategist industry reports, 2020, <https://www.morningstar.com/articles/971254/morningstars-view-the-impact-of-coronavirus-on-the-economy>.
- [3] W. K. R. Fernando, “The economic impact of covid-19,” *Economics in the Time of COVID-19*, vol. 45, p. 45, 2020.
- [4] Q. He, J. Liu, S. Wang, and J. Yu, “The impact of COVID-19 on stock markets,” *Economic and Political Studies*, vol. 8, no. 3, pp. 275–288, 2020.
- [5] A. Saini and A. Sharma, “Predicting the unpredictable: an application of machine learning algorithms in Indian stock market,” *Annals of Data Science*, vol. 6, pp. 1–9, 2019.
- [6] A. Thakkar and K. Chaudhari, “Fusion in stock market prediction: a decade survey on the necessity, recent developments, and potential future directions,” *Information Fusion*, vol. 65, 2021.
- [7] B. Liu, Q. Wu, and Q. Cao, “An improved Elman network for stock price prediction service,” *Security and Communication Networks*, vol. 2020, Article ID 8824430, 9 pages, 2020.
- [8] Y. Chen, K. Liu, Y. Xie, and M. Hu, “Financial trading strategy system based on machine learning,” *Mathematical Problems in Engineering*, vol. 2020, Article ID 3589198, 13 pages, 2020.
- [9] J. Shen and M. O. Shafiq, “Short-term stock market price trend prediction using a comprehensive deep learning system,” *Journal of Big Data*, vol. 7, p. 66, 2020.
- [10] Y. Baydilli and U. Atila, “Understanding effects of hyperparameters on learning: a comparative analysis,” in *International Conference on Advanced Technologies*, Computer Engineering and Science, Ankara, Turkey, 2017.
- [11] P. Probst, A. Boulesteix, and B. B. Tunability, “Importance of hyperparameters of machine learning algorithms,” *Journal of Machine Learning Research*, vol. 20, no. 53, pp. 1–32, 2019.
- [12] Y. Bengio, I. J. Goodfellow, and A. Courville, *Deep Learning*, MIT Press, Cambridge, UK, 2015.
- [13] S. Nitish, H. Geoffrey, K. Alex, S. Ilya, and S. R. Dropout, “A simple way to prevent neural networks from overfitting,” *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [14] S. Hayou, A. Doucet, and J. Rousseau, “On the impact of the activation function on deep neural networks training,” 2019, <http://arxiv.org/abs/1902.06853>.
- [15] L. Smith, “A disciplined approach to neural network hyperparameters: part 1 – learning rate, batch size, momentum, and weight decay,” *CoRR*, vol. 2 2018, <http://arxiv.org/abs/1803.09820>.
- [16] L. Zajmi, H. A. Falah, and A. J. Adam, “Concepts, methods, and performances of particle swarm optimization, back-propagation, and neural networks,” *Applied Computational Intelligence and Soft Computing*, vol. 2018, 7 pages, Article ID 9547212, 2018.
- [17] R. Hecht-Nielsen, “Theory of the backpropagation neural network,” *Neural Networks*, vol. 1, pp. 445–448, 1988.
- [18] I. BU, Z. Baharudin, R. MQ, and P. Nallagownden, “Optimization of neural network architecture using genetic algorithm for load forecasting,” in *Proceedings of the 2014 5th International Conference on Intelligent and Advanced Systems (ICIAS)*, pp. 1–6, Kuala Lumpur, Malaysia, 2014.
- [19] Y. Ghanou and G. Bencheikh, “Architecture optimization and training for the multilayer perceptron using ant system,” *International Journal of Computer Science*, vol. 43, no. 1, p. 10, 2016.
- [20] J. Ananthi and V. Ranganathan, “Multilayer perceptron weight optimization using bee swarm algorithm for mobility prediction,” *IIOAB Journal*, vol. 7, no. 9, pp. 47–63, 2016.
- [21] M. AS and S. YSK, “Web service classification using multi-Layer perceptron optimized with Tabu search,” in *Advance Computing Conference (IACC)*, pp. 290–294, IEEE International, New York, NY, USA, 2015.
- [22] H. Moayedi, D. Tien Bui, M. Gör, B. Pradhan, and A. Jaafari, “The feasibility of three prediction techniques of the artificial neural network, adaptive neuro-fuzzy inference system, and hybrid particle swarm optimization for assessing the safety factor of cohesive slopes,” *ISPRS International Journal of Geo-Information*, vol. 8, no. 9, p. 391, 2019.
- [23] C. C. Aggarwal, “An introduction to neural networks,” in *Neural Networks and Deep Learning*, pp. 1–52, Springer, Cham, Switzerland, 2018.
- [24] R. C. Eberhart and J. Kennedy, “A new optimizer using particle swarm theory,” in *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, pp. 39–43, IEEE, Piscataway, NJ, USA, 1995.
- [25] S. Z. Zhao, J. J. Liang, P. N. Suganthan, and M. F. Tasgetiren, “Dynamic multi-swarm particle swarm optimizer with local search for large scale global optimization,” in *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pp. 3845–3852, Hong Kong, China, 2008.
- [26] C. Hargreaves and Y. Hao, “Prediction of stock performance using analytical techniques,” *Journal of Emerging Technologies in Web Intelligence*, vol. 5, no. 2, pp. 136–142, 2013.
- [27] B. G. Malkiel, *A Random Walk Down Wall Street*, W. W. Norton & Company, New York, NY, USA, 1999.
- [28] T. Helstrom and K. Holmstrom, “Predicting the stock market,” Technical Report IMA-TOM-1997-07, Malardalen University, Sweden, 1997.
- [29] R. W. Colby, *The Encyclopedia of Technical Market Indicators*, McGraw-Hill Press, New York, NY, USA, 2003.
- [30] Historical data for datasets. <https://finance.yahoo.com/world-indices/>.
- [31] R. Arévalo, J. García, F. Guijarro, and A. Peris, “A dynamic trading rule based on filtered flag pattern recognition for stock market price forecasting,” *Expert Systems with Applications*, vol. 81, pp. 177–192, 2017.
- [32] R. Cervelló-Royo, F. Guijarro, and K. Michniuk, “Stock market trading rule based on pattern recognition and technical analysis: forecasting the DJIA index with intraday data,” *Expert Systems with Applications*, vol. 42, no. 14, pp. 5963–5975, 2015.
- [33] T.-l. Chen and F.-y. Chen, “An intelligent pattern recognition model for supporting investment decisions in stock market,” *Information Sciences*, vol. 346–347, pp. 261–274, 2016.
- [34] M.-C. Lee, “Using support vector machine with a hybrid feature selection method to the stock trend prediction,” *Expert Systems with Applications*, vol. 36, no. 8, pp. 10896–10904, 2009.
- [35] R. Cervelló-Royo and F. Guijarro, “Forecasting stock market trend: a comparison of machine learning algorithms,” *Finance, Markets and Valuation*, vol. 6, no. 1, pp. 37–49, 2020.
- [36] M. Nabipour, P. Nayyeri, H. Jabani, and A. Mosavi, “Deep learning for stock market prediction,” <http://arxiv.org/abs/arXiv:2004.01497>.

- [37] H. White, *Economic Prediction Using Neural Networks: The Case of IBM Daily Stock Returns*, Department of Economics, University of California, Los Angeles, CA, USA, 1988.
- [38] N. I. Indera, I. M. Yassin, A. Zabidi, and Z. I. Rizman, "Non-linear autoregressive with exogenous input (narx) bitcoin price prediction model using PSO-optimized parameters and moving average technical indicators," *Journal of Fundamental and Applied Sciences*, vol. 9, no. 3S, p. 791, 2018.
- [39] Y. H. Chen, B. Yang, and A. Abraham, "Flexible neural trees ensemble for stock index modeling," *Neurocomputing*, vol. 70, no. 4-6, pp. 697-703, 2007.
- [40] R. Arjun, "Stock market index prediction using multilayer perceptron and long short-term memory networks: a case study on BSE sensex," *International Journal of Research and Scientific Innovation (IJRSI)*, vol. 5, 2018.
- [41] T. Aboueldahab and M. Fakhreldin, "Stock market indices prediction via hybrid sigmoid diagonal recurrent neural networks and enhanced particle swarm optimization," *International Congress for Global Science and Technology, ICGST*, vol. 10, pp. 23-30, 2010.
- [42] S. R. Das, D. Mishra, and M. Rout, "Stock market prediction using firefly algorithm with evolutionary framework optimized feature reduction for OSELM method," *Expert Systems with Applications: X*, vol. 4, Article ID 100016, 2019.
- [43] Y. Hao and Q. Gao, "Predicting the trend of stock market index using the hybrid neural network based on multiple time scale feature learning," *Applied Sciences*, vol. 10, no. 11, p. 3961, 2020.
- [44] M. Qiu and Y. Song, "Predicting the direction of stock market index movement using an optimized artificial neural network model," *PLoS One*, vol. 11, no. 5, Article ID e0155133, 2016.
- [45] W.-C. Chiang, D. Enke, T. Wu, and R. Wang, "An adaptive stock index trading decision support system," *Expert Systems with Applications*, vol. 59, pp. 195-207, 2016.
- [46] R. A. Jamous, E. El. Seidy, A. A. Tharwat, and B. I. Bayoum, "Modifications of particle swarm optimization techniques and its application on stock market: a survey," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 6, no. 3, 2015.
- [47] C. Worasuchee, S. Nuannimnoi, R. Khamvichit, and P. Attagonwantana, "An automatic stock trading system using particle swarm optimization," in *Proceedings of the 2017 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, pp. 497-500, Phuket, Thailand, 2017.
- [48] R. Ma9jhi, G. Panda, G. Sahoo, A. Panda, and A. Choubey, "Prediction of S&P 500 and DJIA stock indices using particle swarm optimization technique," in *Proceedings of the 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pp. 1276-1282, Hong Kong, China, 2008.
- [49] M. Isiet and M. Gadala, "Sensitivity analysis of control parameters in particle swarm optimization," *Journal of Computational Science*, vol. 41, Article ID 101086, 2020.
- [50] R. Jamous, M. El-Dariby, and H. ALRahhal, "PSO-based artificial neural network configuration selection technique," *Applied Artificial Intelligence Journal*, 2020.
- [51] Y. Liu, Z. Qin, Z. Shi, and J. Lu, "Center particle swarm optimization," *Neurocomputing*, vol. 70, no. 4-6, pp. 672-679, 2007.
- [52] Historical data of Standard's and Poor's 500 (S&P 500). <https://finance.yahoo.com/quote/%5EGSPC/history?p=%5EGSPC>.
- [53] Historical data for DJIA stock market indices. <https://finance.yahoo.com/quote/%5EDJI/history?period1=475804800&period2=1606953600&interval=1d&filter=history&frequency=1d&includeAdjustedClose=true>.
- [54] Standard & Poor's, "Standard & Poor's research insight north America data guide," 2004, <http://140.137.101.73:8008/ri/manual/rium.htm>.
- [55] S&P Dow Jones Indices. Dow Jones industrial average.
- [56] J. Xin, G. Chen, and Y. Hai, "A particle swarm optimizer with multi-stage linearly-decreasing inertia weight," in *Proceedings of the 2009 International Joint Conference on Computational Sciences and Optimization*, pp. 505-508, Sanya, Hainan, 2009.
- [57] S. Sengupta, S. Basak, and R. A. Peters, "Particle swarm optimization: a survey of historical and recent developments with hybridization perspectives," *International Journal of Computer Theory and Engineering*, vol. 1, pp. 157-191, 2019.
- [58] G. Panchal, A. Ganatra, and Y. Kosta, "Behaviour analysis of multilayer perceptrons with multiple hidden neurons and hidden layers," *International Journal of Computer Theory and Engineering*, vol. 3, no. 2, 2011.
- [59] A. J. Thomas, M. Petridis, S. D. Walters, S. M. Gheytaasi, and R. E. Morgan, "On predicting the optimal number of hidden nodes," in *Proceedings of the 2015 International Conference on Computational Science and Computational Intelligence*, pp. 565-570, Las Vegas, CA, USA, 2015.
- [60] A. J. Thomas, S. D. Walters, S. D. Walters, S. M. Gheytaasi, R. E. Morgan, and M. Petridis, "On the optimal node ratio between hidden layers: a probabilistic study," *International Journal of Machine Learning and Computing*, vol. 6, no. 5, pp. 241-247, 2016.
- [61] F. Ye, "Particle swarm optimization-based automatic parameter selection for deep neural networks and its applications in large-scale and high-dimensional data," *PLoS One*, vol. 12, no. 12, Article ID e0188746, 2017.
- [62] J. J. Moré, "The Levenberg-Marquardt algorithm: implementation and theory," in *Lecture Notes in Mathematics*, G. A. Watson, Ed., vol. 360, pp. 105-116, Springer-Verlag, Berlin, Germany, 1978.
- [63] A. Botchkarev, "A new typology design of performance metrics to measure errors in machine learning regression algorithms," *Interdisciplinary Journal of Information, Knowledge, and Management*, vol. 14, pp. 045-076, 2019.
- [64] M. Stang, C. Meier, V. Rau, and E. Sax, "An evolutionary approach to hyper-parameter optimization of neural networks," in *Human Interaction and Emerging Technologies. IHET 2019*, T. Ahram, R. Taiar, S. Colson, and A. Choplin, Eds., vol. 1018, Springer, Cham, Switzerland, 2020.
- [65] The drop of the S&P 500 index in USA during the COVID-19 crisis report. <https://voxeu.org/article/stock-market-and-economy-insights-covid-19-crisis>.
- [66] WHO coronavirus disease (COVID-19) dashboard. <https://covid19.who.int/>.
- [67] Historical data for stock market indices. <https://finance.yahoo.com/lookup>.