mathematics

Article

An Evolutionary Technique for Building Neural Network Models for Predicting Metal Prices

Devendra Joshi [1], Premkumar Chithaluru [2,3,4,*], Divya Anand [3,5,6], Fahima Hajjej [7], Kapil Aggarwal [1], Vanessa Yelamos Torres [6,8,9] and Ernesto Bautista Thompson [3,6,8]

[1] Department of CSE, Koneru Lakshmaiah Education Foundation, Guntur 522302, Andhra Pradesh, India
[2] Department of Computer Science and Engineering, Chaitanya Bharathi Institute of Technology, Hyderabad 500075, Telangana, India
[3] Department of Project Management, Universidad Internacional Iberoamericana, Campeche 24560, Mexico
[4] Uttaranchal Institute of Technology, Uttaranchal University, Dehradun 248007, Uttarakhand, India
[5] Department of Computer Science and Engineering, Lovely Professional University, Phagwara 144411, Punjab, India
[6] Higher Polytechnic School, Universidad Europea del Atlántico, C/Isabel Torres 21, 39011 Santander, Spain
[7] Department of Information Systems, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia
[8] Department of Project Management, Universidad Internacional Iberoamericana, Arecibo, PR 00613, USA
[9] Engineering Research and Innovation Group, Universidade Internacional do Cuanza, Estrada Nacional 250, Bairro Kaluapanda, Cuito EN250, Angola
* Correspondence: bharathkumar30@gmail.com or premkumarc_cse@cbit.ac.in

Abstract: In this research, a neural network (NN) model for metal price forecasting based on an evolutionary approach is proposed. Both the neural network model's network parameters and network architecture are selected automatically. The time series metal price data set is used to construct a novel fitness function that takes into account both error minimizations and the reproduction of the auto-correlation function. Calculating the average entropy values allowed the selection of the input parameter count for the neural network model. Gold price forecasting was performed using the proposed methodology. The optimal hidden node number, learning rate, and momentum are 9, 0.026, and 0.76, respectively, according to the evolutionary-based NN model. The proposed strategy is shown to reduce estimation error while also reproducing the auto-correlation function of the time series data set by the validation results with gold price data. The performance of the proposed method is better than other current methods, according to a comparison study.

Keywords: evolutionary algorithm; auto-correlation; metal price; neural network; cross-validation; entropy

MSC: 58Z99

check for updates

Citation: Joshi, D.; Chithaluru, P.; Anand, D.; Hajjej, F.; Aggarwal, K.; Torres, V.Y.; Thompson, E.B. An Evolutionary Technique for Building Neural Network Models for Predicting Metal Prices. *Mathematics* **2023**, *11*, 1675. https://doi.org/10.3390/math11071675

Received: 25 February 2023
Revised: 22 March 2023
Accepted: 23 March 2023
Published: 31 March 2023

1. Introduction

Mineral pricing has a big impact on how minerals are produced and how different strategic selections are made in the mineral sectors. The price of minerals plays a significant effect in our daily lives. Additionally, this pricing is crucial for investments in commodities, project evaluations, and strategic planning. They also reflect and have an impact on overall economic activity. Numerous factors have a big impact on the prices of minerals and the mineral market. However, it is quite challenging to understand how those components work and how they affect mineral prices, which makes investing in the mineral market extremely risky. Moreover, mining capital investments are also particularly uncertain due to the fluctuation of the mineral market and the pricing of mineral commodities. As a result, when making decisions, it is crucial to be able to predict the price of mineral resources accurately.

Lifetime distribution models and Grey–Markov models are the foundations of conventional price estimation techniques [1]. Due to various a priori assumptions, these techniques have some limitations. It is challenging to verify these assumptions given the current prices for mineral commodities. As a result, the sustainability of these models is always uncertain. The pricing models for mineral commodities are also made more complex and nonlinear by the fact that a variety of factors affect mineral prices [2]. The results from conventional commodity price models are not sufficient [3].

Numerous researchers have worked hard to advance forecasting approaches by putting forward a variety of strategies and models to get around the limitations of traditional price prediction models [4–8]. Linear statistical models are used as part of the autoregressive moving average (ARIMA) methodology, as introduced by Box and Jenkins [9]. For predicting metal prices, various academics have developed different variants of ARIMA models [10–12]. The formal model formulation required with an assumed probability distribution for the data is the main disadvantage of ARIMA-based models. To solve this issue, McDonald and Xu [13] suggested partially adaptive estimate methods that make use of extremely flexible families of distributions. However, their method is only applicable to specific families of distribution functions.

For price forecasting models, neural networks (NNs) are becoming more popular than ARIMA models due to their robustness when processing non-normal and nonlinear data [14]. NNs are incredibly adaptable and do not need formal model specifications or a presumption about how the data will be distributed in probability [15]. Additionally, NNs handle chaotic components with extremely thick tails better than most alternative approaches [16]. There is limited scope for ANN applications in forecasting mineral prices. To forecast the price of gold, Chunmei and Lian et al. [14,15] used a back propagation neural network technique. A neuro-fuzzy algorithm was employed by Shijiao et al. [17] to predict mineral prices. There have been a lot of studies conducted using these techniques and other forms of time series forecasting data sets, although the use of soft computing, including neural networks, is limited for forecasting mineral prices [18–27].

To enhance the reliability of neural network forecasting models, many investigators have established a range of hybrid techniques. Wedding and Cios [28] demonstrated a combination network that included radial basis function networks and the Box–Jenkins model. Armano et al.'s [29] unique hybrid strategy combining genetic algorithms (GAs) with artificial neural networks was proposed as a way to forecast the stock market (ANNs). Various ensembles of neural network models have also been presented by researchers to increase the accuracy of forecasting systems [30,31]. For forecasting drug dissolution profiles, Goh et al. [32] employed an ensemble of boosted Elman networks. Using generalized linear autoregression (GLAR) and artificial neural networks, Yu et al. [33] introduced a unique nonlinear ensemble forecasting model to obtain precise predictions of the foreign exchange market.

The most often used architecture for time series forecasting is multilayer perceptron neural networks [34–36]. The difficulty of accurately fitting the underlying data distribution is the primary drawback of multilayer perceptron neural networks [37]. The incorrect selection of neural learning parameters, such as learning rate, momentum, and incorrect network design, such as hidden node size and a number of hidden layers, frequently leads to fitting issues. Therefore, designing the optimal forecasting model by appropriately choosing learning parameters and network architecture is a key issue. The most widely used strategy for design is the trial-and-error heuristic for choosing learning parameters and network architecture; however, this method does not always result in the best prediction. The trial-and-error heuristic methods also require longer computing times.

An automatic design approach can be used as an alternative to heuristic methods for choosing the neural network parameters. It has been demonstrated that using an evolutionary algorithm instead of heuristic methods to find the optimal ANN model is preferable [38]. To provide more adaptable neural network modeling, some academics have proposed evolutionary computation for evolving sparsely connected topologies [18,21,39].

However, the evolutionary computation-based ANN techniques that have been published so far have taken into account error reduction is a crucial criterion for choosing the model hyperparameters [40]. However, there is no assurance that the projected values of such a model will accurately reproduce the temporal correlation structure of time series data [41]. This might provide a smoothing effect in the model, which would overestimate the low-valued data and underestimate the high-valued data while failing to reproduce the data variability. With the help of the auto-correlation function, the correlation structure of the time series data is quantified.

Determining a flexible structure for a metal price forecasting model using an evolutionary-based ANN model was the objective of this research. Based on the average entropy values, the ANN model's input variable selection was made [42–47]. The optimal hidden layer node count and neural network learning parameters are selected using the evolutionary approach. The fitness function in this research added a new term that guarantees the replication of the auto-correlation function of the metal price data, as opposed to previous work that used error minimization as the fitness function.

This research included the following contributions:

- an evolutionary-based ANN model to generate flexible structure for a metal price forecasting model was developed;
- ANN input parameters variable selection was based on average entropy values;
- the evolutionary algorithm is used for selecting optimum hidden nodes number in the hidden layer and neural network learning parameters;
- this work included a new component to the fitness function that ensures the reproduction of the auto-correlation function of metal price data.

The structure of the article is as followed by the literature survey in Section 2. The suggested approach for forecasting metal prices is described in Section 3. In Section 3.1, a brief description of time series analysis application to metal price modeling is provided, and Section 3.2 outlines the process for choosing the best-lagged data. Section 3.3 presents the neural network modeling for metal price forecasting, while Section 3.4 discusses the evolutionary approach used in the neural network model creation. In Section 4, a forecasting case study for the price of gold is given, in continuation with result and discussion and a set of conclusions.

2. Literature Survey

Throughout the past few decades, numerous researchers have analyzed studies about gold prices and the variables that affect their fluctuations. This subject is still quite popular in research on global economic and financial issues. According to three primary categories, studies on the factors that influence the price of gold can be categorized.

The first strategy incorporates forecasting price trend by analyzing the fluctuation of the gold price in terms of previous prices. In order to predict the price of gold bullion coins from 2002 to 2007, Abdullah [48] built an ARIMA model. The findings suggest that the optimal model to use is ARIMA (2, 1, 2). According to Khan's [49] ARIMA forecasting model, which covered the years 2003–2012, The most appropriate model to use is ARIMA (0,1,1). A model called ARIMA was also created to estimate gold prices for the years 2003–2012 [50]. They discovered that this model is the most accurate (7,1,10). From 2003 to 2014, Guha and Bandyopadhyay projected the price of gold in India [24]. The findings indicate that (1, 1, 1) ARIMA is the preferred model to forecast future gold prices. However, this method is only employed in the short term. Tripathy [51] utilized the ARIMA model to predict India's gold price from 1990 to 2015, and the results indicate that ARIMA (0,1,1) is the best model to use.

In second strategy, bivariate and multivariate analysis of the variance in the key macroeconomic factors is used to model the fluctuations in gold prices. According to Šimáková [52], who looked into the link between oil and gold prices from 1970 to 2010, causal relationships between oil and gold price levels were found using the Granger causality test, and a long-term connection among oil and gold was disclosed using the Johansen co-integration test. However, for analyzing the short-term variance in co-integrated time series EC model, CPI is used but it is not analyzed in its terms as GMI (gold mining index) and VEC model is stated. The capacity of gold prices to forecast the AuAUDexchange rate relative to the USD exchange rate was studied [53]. The results, which were obtained using an EC model using data from 2000 to 2012, demonstrate the existence of co-integration between AUD/USD exchange rate, with the coefficient on gold prices being appropriately signed and statically relevant. The economic effects on gold prices from 1994 to 1997 were examined [54]. They discovered that the return volatility of the gold market is significantly influenced by employment data, GDP, CPI, and personal income using the fractionally integrated GARCH (FIGARCH) model and flexible Fourier form (FFF) regression. They also mentioned the longer history characteristics of the gold market price fluctuations. A conceptual framework was created by Levin and Wright to investigate the factors that affected the price of gold in the short- and long-term from 1976 to 2005 [55]. They discovered an enduring connection between the gold price and the level of the U.S. dollar using co-integration regression techniques. In the short run, there was a statistically significant positive association among changes in the price of gold and variations in U.S. consumer price index, U.S. inflation currency fluctuations, and credit risk, despite the fact that there was a statistically significant negative association between variations in the price of gold and changes in the USD investment, rate of exchange and the gold rental price. The third strategy focuses on simulating changes in macro-financial variables, including speculation about gold price movements and economic indexes, in order to predict how gold prices will move in the future. Rates of both the gold lease and the exchange. Regression analysis was used by Baker and Van Tassel to create a model that could predict changes in the price of gold [56]. The model's output demonstrated that variations in the gold price can be explained by variations in the value of commodities, USD, the dollar's value, and the rate of future inflation. Additionally, asset bubbles were substantial with positive coefficients, lending credence to the idea that speculation pushed the price of gold beyond its trend. The link between gold and financial indicators from 1975 to 2001 was examined by Lawrence using a VAR model. According to the findings, there is no statistically significant relationship between gold returns and changes in macroeconomic variables such as GDP, unemployment, and rate of interest, even though these changes have a considerably greater impact on other commodities than gold [57]. Tully and Lucey [58] looked into the economic factors that affected the price of gold between 1983 and 2003. The findings of the VAR analysis demonstrate that the U.S. dollar is the primary factor influencing the gold price, which is affected by the FTSE cash, dollar, pound, and U.S. rate of interest as well as the UK consumer price index.

3. Proposed Methodology

The Proposed methodology is divided into 4 parts: Section 3.1—time series analysis is used to model metal price; Section 3.2—optimum lagged data selection; Section 3.3—metal price prediction using a neural network method; Section 3.4—neural network with an evolutionary algorithm for metal price modeling.

3.1. Time Series Analysis Is Used to Model Metal Price

The use of time series analysis in metal price modeling is regarded as a data-driven model where the relationship between historical past price data and future price data is obtained by an appropriate training procedure. The future metal price is then predicted using the relationship that has been formed. The inputs used by the time series model for univariate time-series analysis forecasting metal prices are the previous observations of the metal prices, and the outputs are the future price.

If $X = [x_1, x_2, \ldots, x_t]$ consists of metal price time series data and the time-series model has p input variables $t = (p + i + 1), i \in (0, 1, \ldots, t - p - 1)$; therefore it is possible to extract the pattern data set from X in the following manner to create the metal price model:

$$T_i = \left\{ x_{(1+i)},\ x_{(2+i)}, \ldots, x_{(k+i)}, \ldots,\ x_{(p+i),}\ x_{(p+i+1)} \right\},\quad i\,\epsilon\,\{0,1,2,\ldots,t-p-1\},\ k < p \tag{1}$$

Time series forecasting with NN as instead of NN is used in the metal price modeling is defined as establishing the relationship based on the value at the period $x_{(p+i+1)}$ and the data from the time series' previous elements, employing many lags [59] $\left\{ x_{(1+i)}, x_{(2+i)}, \ldots, x_{(p+i)} \right\}$, to obtain a function as:

$$x_{(p+i+1)} = f\left(x_{(1+i)}, x_{(2+i)}, \ldots, x_{(p+i)} \right), i \in \{0,1,2,\ldots,t-p-1\} \tag{2}$$

where

$\left\{ x_{(1+i)}, x_{(2+i)}, \ldots, x_{(p+i)} \right\}$ is a time-lagged pricing vector;

$x_{(p+i+1)}$ is the price at time t

p is the number of previous metal price observations associated with the future value.

This work aims to use the NN model to approximate the function f of Equation (2).

Before developing the NN model to forecast metal price values, an initial step, i.e., normalizing the data has to be performed with the observed price values. The original values for x_i are normalized to N_i within the range [0, 1] using the following equation:

$$N_i = \frac{(x_i - x_{min})}{(x_{max} - x_{min})} \tag{3}$$

where

$$\widetilde{x}_{(p+1+i)} = x_{min} + \widetilde{N}_{(p+1+i)} * (x_{max} - x_{min})$$

The NN model of metal price forecasting was developed using the normalized data, and once the NN outputs the resulting values of $\widetilde{N}_{(p+1+i)}$, rescaling was performed using the below equation to obtain the original scale data $\widetilde{x}_{(p+1+i)}$:

$$\widetilde{x}_{(p+1+i)} = x_{min} + \widetilde{N}_{(p+1+i)} * (x_{max} - x_{min}) \tag{4}$$

where,

$\widetilde{N}_{(p+1+i)}$ is NN model output value, and $\widetilde{x}_{(p+1+i)}$ is the rescaled price value at $t = (p+i+1), i \in (0,1,\ldots,t-p-1)$.

In this work, a one-step-ahead forecast model was created, which only needs one neuron at the output layer; however, multi-step-ahead forecasts can also be created by iteratively employing one-step-ahead forecasts as inputs [18].

To develop the NN model for metal price forecasting, the pattern data set extracted in Equation (1) from the metal price data $x = [x_1, x_2, \ldots, x_t]$ was normalized using Equation (3). From Equation (2), observe that the total $(t-p)$ numbers of patterns set can be extracted from X, where

p = number of lagged variables

t = no. of data observed in the metal price data set X.

3.2. *Optimum Lagged Data Selection*

It is obvious from Section 3.1 that selecting lagged data p is the most important step in time series analysis-based metal price modeling. The size of p should be chosen so that it captures both the large-scale variability and the small-scale features that are present in the metal price data. To optimize small-scale features, the value of p selected is small; to capture the true features included in the defined time series data, it should, however, be chosen as broadly as necessary [47].

Entropy is a popular method for selecting patterns from one-dimensional signals or multi-dimensional images [46,47]. Throughout this study, the optimum lagged size p was determined using the average entropy value of all created patterns. According to [60], the average entropy of the patterns set T_i with the dimensions of $(p+1)$ can be calculated as follows:

$$H = \frac{1}{K}\sum_{i=1}^{K} q_i log(q_i) \tag{5}$$

where

K = all possible outcomes randomly generated.
q_i = mass function probability.

3.3. Metal Price Prediction Using a Neural Network Method

The NN technique was applied to create the metal price forecasting model [20,37]. Multilayer neural networks (MNN), an important part of perceptron neural networks and applied widely in many fields, were applied in this paper. This method is capable of achieving pattern recognition by providing an approximate expression for an objective function. The NN model's input nodes (variables) are $\left\{N_{(1+i)}, N_{(2+i)}, \ldots, N_{(p+i)}\right\}$, and the variable output node is $N_{(p+i+1)}$. There was no specified value assigned to the number of nodes in the hidden layer. Instead, it was chosen using the suggested algorithm that was presented in this work. These nodes are linked together via connections of varying strength. In need of a node to function, each incoming signal is multiplied by a weight before the weighted inputs are summed. For NN models, this study used the gradient descent method combined with momentum learning. The continuous process of choosing appropriate weights minimizes the objective function:

$$E(w) \equiv \frac{1}{2*c}\sum_{i=1}^{c}\left|(N_{(p+i+1)} - \widetilde{N}_{(p+i+1)})\right|^2 \tag{6}$$

where

$N_{(p+i+1)}$ and $\widetilde{N}_{(p+i+1)}$ = targets normalized,
c = length of network output;
w = network's weights (the weights are initially set to random values before being modified to reduce the error).

$$\Delta w(m) = -\eta\frac{\partial E(m)}{\partial w} + \mu\Delta w(m-1) \tag{7}$$

where,

η is the rate of learning, and μ is momentum.

At iteration m of the iterative method, a weight vector must be taken and updated as $w(m+1) = w(m) + \Delta w(m)$. By using Equation (6) as an objective function, one aims to reduce the variances between observed values and those predicted by the model. However, there is no guarantee that values predicted by minimizing Equation (6) would replicate the auto-correlation structure of time series data. The auto-correlation of time series data was taken into account in this study along with a new objective function that takes advantage of this property of MNN by reducing the objection of Equation (6).

$$E_{new}(w) \equiv \frac{1}{2}\{\frac{1}{c}\sum_{i=1}^{c}\left|(\frac{N_{(p+i+1)} - \widetilde{N}_{(p+i+1)}}{\overline{N}})\right|^2 + \frac{1}{m}\sum_{v=1}^{m}(\frac{C(h_v) - C^*(h_v)}{\overline{C}})^2\} \tag{8}$$

where $C(h_v)$ and $C^*(h_v)$ are auto-correlation constructed from sample data and estimation values from the network, respectively; h_v is the distance of the vth data pair; m is the number of lags used for the calculation, and $\overline{N}$ and $\overline{C}$ are averages of $N_{(p+i+1)}$ and $C(h_v)$, respectively.

Every time the error function of Equation (8) iterates, the weight updating process is carried out until the error hits the threshold point. The discussion above makes it clear these three parameters (learning rate (η), momentum (μ), and number of hidden nodes in the hidden layer (H_n)) are crucial factors that must be chosen to ensure that the network learns properly. In the past, these parameters have been chosen through trial and error, which results in a poor network model. The parameters for NN learning were selected using an evolutionary method.

3.4. Neural Network with an Evolutionary Algorithm for Metal Price Modeling

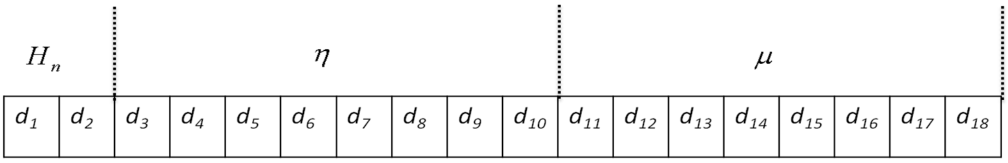
The process of designing an NN model involves searching through all potential NN models. While many evolutionary algorithms can be employed for this search, the genetic algorithm technique was chosen for this work because of its straightforward structure and easily implementable features [61–63].

The algorithm starts with a population of chromosome-based solutions and then modifies itself using 3 genetic parameters (selection, crossover, and mutation) to produce a new generation of solutions that are better than the previous one. To figure out the probability that a specific chromosome will be passed down to the following generation, the fitness function is used. Some chromosomes are chosen for the crossover process whereas others are chosen by elitism based on personal fitness values. The parent solution is a collection of chromosomes chosen for crossover. A pair of parent solutions produces two child solutions as a result of crossover. According to a user-selected mutation rate, mutation often modifies the binary value, moving it randomly from 0 to 1 or 1 to 0, depending on the case. The algorithm can get away from local minima with the help of the mutation operation. To maintain population size, the population's worst solutions are eliminated once the three genetic procedures, which provide fitness values, are completed. A generation refers to this single operational cycle. After one generation, the obtained chromosome population will serve as the starting point for the upcoming generation. This process keeps on until it hits the threshold value or predetermined generation number. The primary goal of this evolutionary method is to identify the global minimum zone. A local optimization technique called the gradient descent algorithm looks for the best answer inside the minimum zone. A fully linked multilayer perceptron with just one hidden layer and one output node was employed. The suggested method entails numerous initializations and development of the NN topology, including the hidden node size and learning parameters. The optimized parameters include the number of hidden neurons (H_n) of the hidden layer, learning rate (η), and momentum (μ). A direct encoding schema, which is placed into the chromosome to represent one random solution for the parameters, was used in this paper. The chromosomes represented all three parameters: number of hidden neurons (H_n), learning rate (η), and momentum (μ). Two decimal digits (i.e., from 0 to 9) were used to codify the number of hidden nodes (H_n), whereas a binary coded number was used for coding learning rate (η) and momentum (μ). A binary code number of 8 bits was used to indicate both the learning rate (η) and momentum (μ) parameters. As a result, each chromosome has 18 bits, 2 for hidden node number, and 8 for each learning parameter. Figure 1 represents an encoded chromosome of the neural network parameters. The first two digits represent the hidden node size and can take any integer value between 1 to 100.

Figure 1 Learning encoding strategy for neural networks based on evolutionary algorithms Eight bits for the parameter learning rate (η) can only code numbers 0 to 255. If this representation is divided by 255, then one obtains a coding for the possible η values lying within [0, 1]. In the same way, the momentum (μ) parameter was also represented by 8 bits and divided by 255 to keep that parameter value within [0, 1]. It is noted that η and μ can take any real values; however, for searching within a reasonable time, these values are considered. For example, 09 10011011 01101011 is a single chromosome that represents that the value of a hidden node is 9, the learning rate is 0.608, and the momentum is 0.42.

The initialization of the chromosome was uniform. The population size, or total number of chromosomes in the population, was 50. Based on an individual's fitness, a probabilistic selection is made with the better prospects having a higher chance of being chosen. The normalized geometric ranking system used in this study $p_i = q'(1-q)^r$ was applied.

where
p_i = probability that individual i^{th} will be chosen;
q = probability of choosing the ideal candidate;
r = individual rank;
p = population size.



$$d_i \in \{0,1,\ldots,9\} \text{ if } i = 1 \text{ or } 2 \quad \textit{else}$$

$$d_i \in \{0,1\}$$

$$H_n = d_1 * 10 + d_2 + 1$$

$$\eta = \frac{d_3 * 2^7 + d_4 * 2^6 + d_5 * 2^5 + d_6 * 2^4 + d_7 * 2^3 + d_8 * 2^2 + d_9 * 2^1 + d_{10} * 2^0}{2^8 - 1}$$

$$\mu = \frac{d_{11} * 2^7 + d_{12} * 2^6 + d_{13} * 2^5 + d_{14} * 2^4 + d_{15} * 2^3 + d_{16} * 2^2 + d_{17} * 2^1 + d_{18} * 2^0}{2^8 - 1}$$

Figure 1. Learning encoding strategy for neural networks based on evolutionary algorithms.

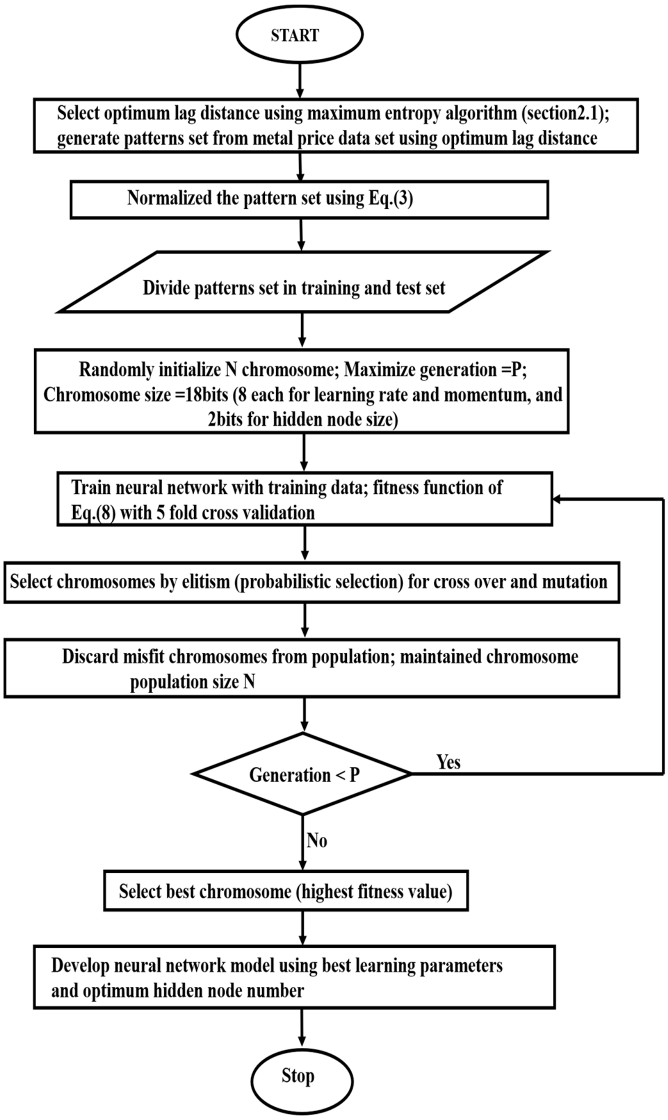
To create a superior solution from the available solutions, each generation undergoes the crossover process. This process includes rearranging the genetic material on the chromosomes to aid individuals who can gain from their parents' fitness. This study employed a uniform crossover with a 0.1 probability rate. The genetic operator known as mutation keeps population diversity preserved. By probabilistically "flipping" portions of the chromosome at random, mutation works. Assuming that p is the size of each of the two chromosome sections, the mutation probability used in this study was $1/p$. Random immigration increases population diversity and lessens the probability of early convergence [64]. A similar number of newly initialized random individuals are added to the population to replace those with low fitness levels. In this study, the number of individuals removed as well as the number of newly introduced individuals in each generation equaled 5. Equation (8) was employed as the fitness function for evolutionary algorithms on the validation data set. To increase the generalization abilities of the created NN model, the k-fold cross-validation method was used. The fitness function can be expressed as the average of the k-fold cross-validation data.

$$\text{Fitness function} = \frac{1}{k}\sum\nolimits_{j=1}^{k}\left\{\frac{1}{n}\sum\nolimits_{i=1}^{n}\left|\left(\frac{N_{(p+i+1),j} - \widetilde{N}_{(p+i+1),j}}{\overline{N}}\right)\right|^2 + \frac{1}{m}\sum\nolimits_{v=1}^{m}\left(\frac{C(h_{v,j}) - C^*(h_{v,j})}{\overline{C(j)}}\right)^2\right\} \tag{9}$$

where,

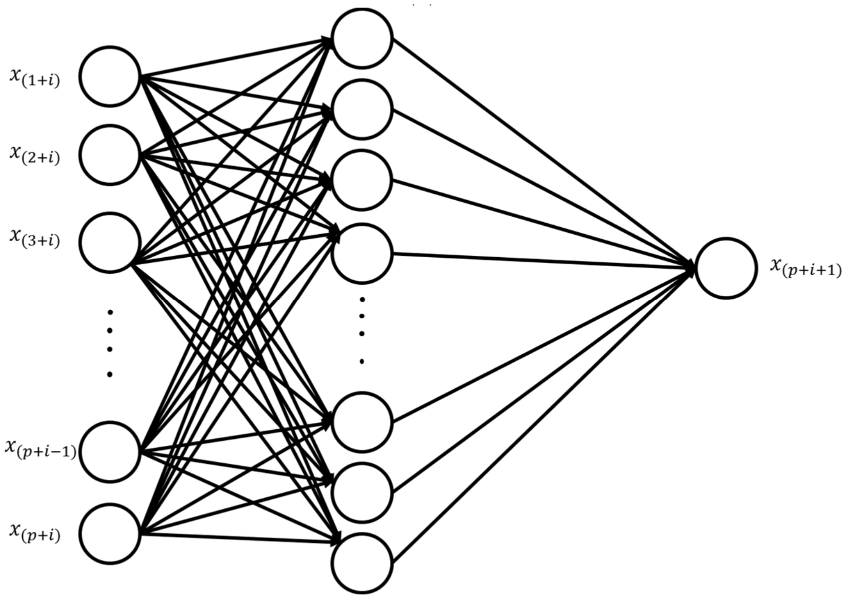
$N_{(p+i+1),j}$ and $\widetilde{N}_{(p+i+1),j}$ are the normalized value, and the network predicted values, respectively, at the jth fold. n = represents the number of validation data at the jth fold. $C(h_{v,j})$ and $C^*(h_{v,j})$ are auto-correlations constructed from sample data and estimation values from the network at the jth fold, respectively.

The fitness function was used to assess the fitness ratings of all these chromosomes. Elitism chose some of the chromosomes. Chromosomes were chosen for crossover and mutation operations using the probabilistic-selection criterion. To maintain a constant population size, many chromosomes that fit poorly were removed from the chromosome solution. The original population size was 50. Poorly fitted solutions were removed after each generation to retain a population size of 20. Each chromosome's fitness value, which illustrates the evaluation of its performance using testing data, has been calculated using training data. It is more likely that the chromosome with the highest fitness function value will be chosen for the following GA generation. The study made use of a reproduction roulette wheel operator. Based on fitness criteria, elitism selects particular chromosomes. This genetic method was repeated until a generation value of 50 was obtained. The model returned a collection of final solutions after achieving maximal generation (50). The optimum solution for the model is the chromosome with the lowest error value. For the NN model, that chromosome has the ideal learning parameters (learning rate and momentum). It should be noted that in this work, hidden layer nodes are not taken into account as learning parameters. There was just one hidden layer network deployed. Figure 2a summarizes the evolutionary-based neural network approach for metal price forecasting and Figure 2b shows the structure of the neural network used.


START
Select optimum lag distance using maximum entropy algorithm (section2.1); generate patterns set from metal price data set using optimum lag distance
Normalized the pattern set using Eq.(3)
Divide patterns set in training and test set
Randomly initialize N chromosome; Maximize generation =P; Chromosome size =18bits (8 each for learning rate and momentum, and 2bits for hidden node size)
Train neural network with training data; fitness function of Eq.(8) with 5 fold cross validation
Select chromosomes by elitism (probabilistic selection) for cross over and mutation
Discard misfit chromosomes from population; maintained chromosome population size N
Generation < P
Yes
No
Select best chromosome (highest fitness value)
Develop neural network model using best learning parameters and optimum hidden node number
Stop

(**a**)

Figure 2. *Cont.*



(**b**)

Figure 2. (**a**) The proposed algorithm was used in this study. (**b**) The structure of the neural network used in this study.

4. Case Study for Gold Price Forecasting Application

The proposed method was used to create a gold price forecasting model. The data are from the index mundi commodity price index data collection (http://www.indexmundi.com accessed on 18 December 2022). This study used monthly gold price data from January 1990 through December 2020. As a result, this study has 360 data points in total. Figure 3 depicts time series gold price data. All tests were carried out on Intel(R) 4.2 GHz Core i5 processor with 8 GB of RAM. The python platform was used to perform all ANN and GA procedures. The code was created by merging the GENLAG [63,65] and AMORE [36,66] tools.

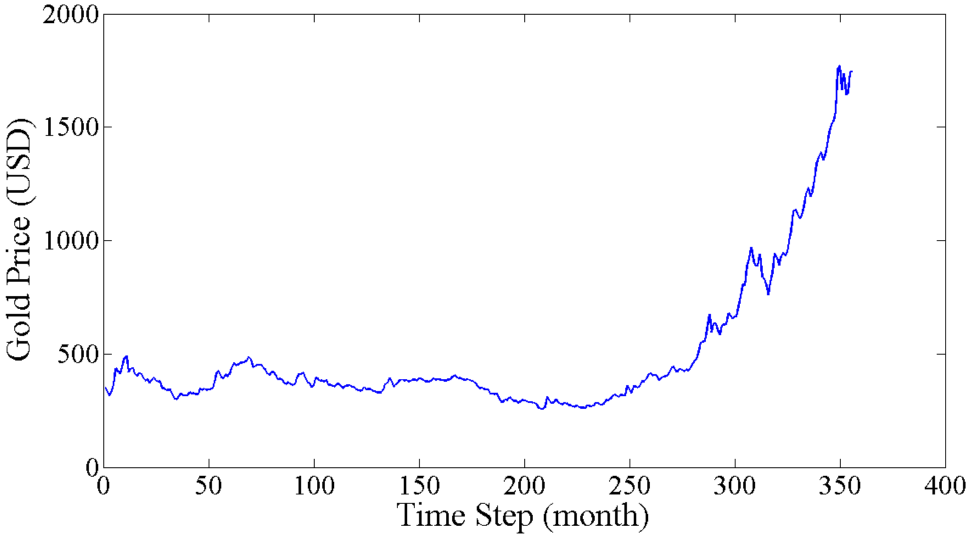


Figure 3. The study’s data set for gold prices.

The evolutionary-based NN model for gold price forecasting is made up of input, hidden, and output nodes. The monthly gold price data set’s patterns data were first extracted using Equation (1) after choosing the lagged value p to create the NN model. The lagged value was chosen by calculating the average entropy of all potential lagged values, as was covered in Section 3.2. Figure 4 shows the average entropy values of the gold price data set with lagged values. The optimum lagged value, as determined by the average entropy values, is 5.

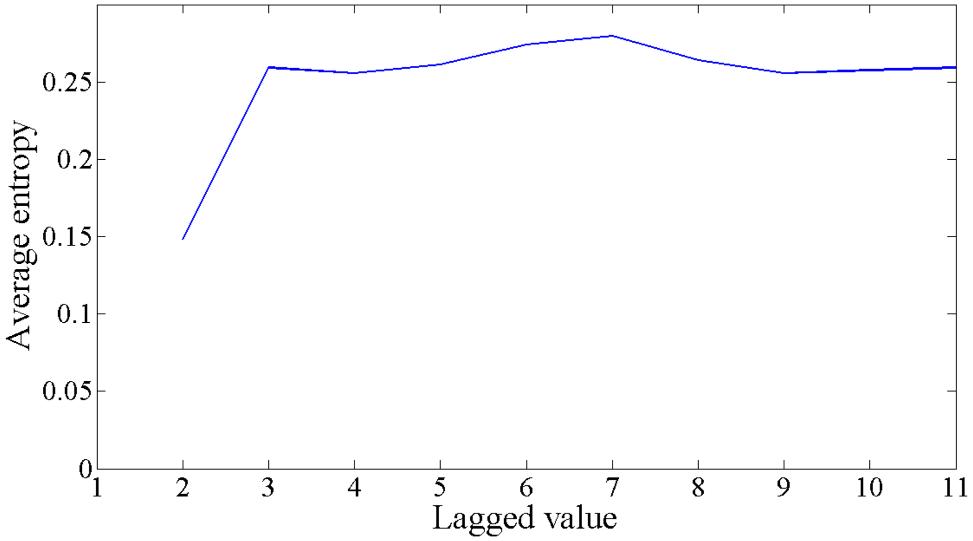


Figure 4. Values of mean entropy for various T_i size.

After choosing the p-value of 5, a total of 356 pattern sets were produced using monthly gold price data. Initially, 284 patterns (about 80% of all patterns) were utilized as training data for the creation of NN models, while the remaining 72 patterns (roughly 20% of all patterns) were used for testing. The normalizing function of Equation (3) was used to scale each pattern between 0 and 1. The adaptive gradient descent using the momentum learning algorithm was utilized for NN modeling, and the tan-hyperbolic and pure linear activation functions were applied in the hidden and output layers, respectively. For evolutionary learning, an 18-bit chromosome was employed. The first two bits represent the hidden node number. The following 8 bits indicate the learning rate, and the last 8 bits indicate the momentum parameter. A total of 50 such chromosomes were produced at random to represent the initial parameter populations. With these parameter populations, NN models were created. Populations of parameters were ranked based on the fitness function. Equation (9) was applied to the data set with k-fold cross-validation as a fitness function. The value of k in the k-fold cross-validation study was selected as 5. Then, the evolution-based learning determined the optimum values of the hidden node number (H_n), learning rate (η), and momentum (μ) as 9, 0.026, and 0.76, respectively.

The optimal parameters were then used to create the neural network model. Figure 5 depicts a scatter plot of actual and predicted gold price training data values, demonstrating that predicted values agree well with actual values. To evaluate the suggested method's capacity to reproduce the metal price data set's auto-correlation, auto-correlation values up to a lag of 50 were calculated for both the actual and estimated gold price training data sets (Figure 6). The real auto-correlation function completely matched the estimated auto-correlation function using the suggested evolutionary-based NN model, as shown in Figure 6.

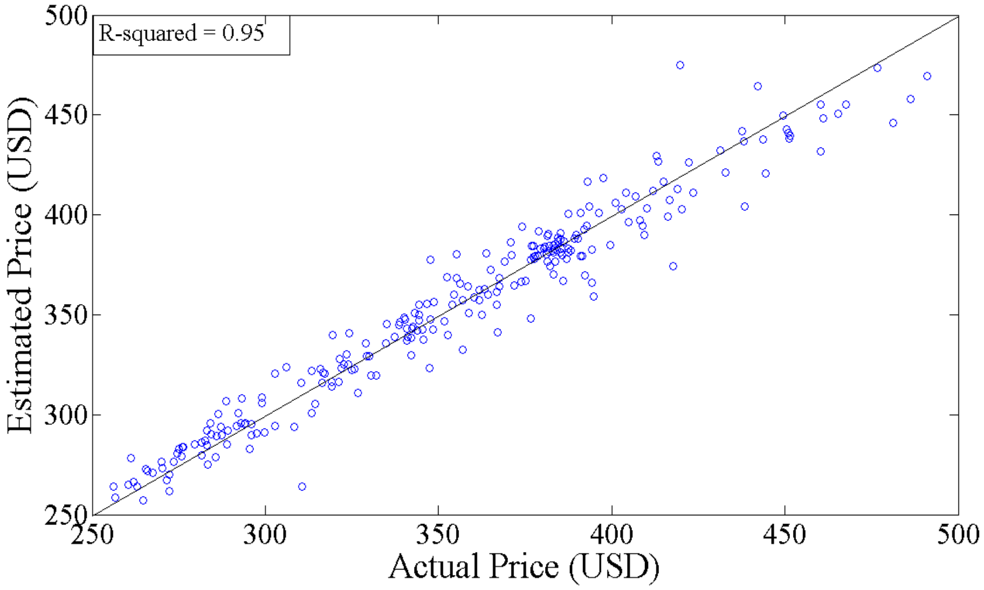


Figure 5. Training data for actual and forecast gold prices.

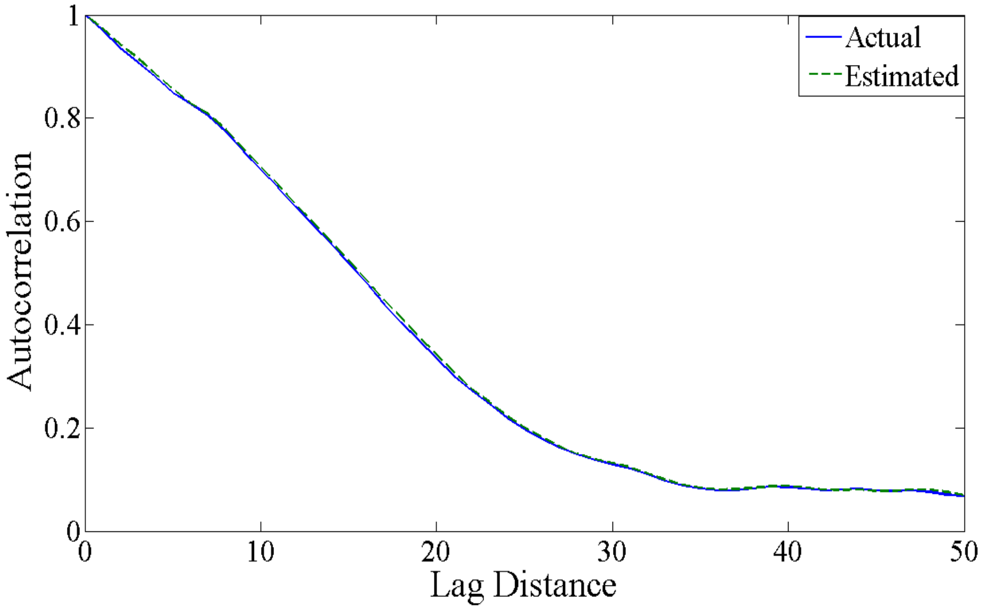


Figure 6. Actual and predicted gold price auto-correlation plots for the training data set.

5. Result and Discussion

To validate the suggested strategy, the generalization ability and performance of the evolutionary-based NN model were applied to data that had not previously been utilized for training. This validation was carried out using the test data set. The developed NN model was tested on data. The anticipated values were calculated and then compared to their actual values. The errors were determined as the difference between the actual and predicted values from the model. The estimated errors were used to generate error statistics such as the mean error, mean absolute errors, mean squared error, error variance, and coefficient of determination (R^2). Table 1 displays the error statistics for the actual values and model anticipated values for the training and test data sets. For the training and test data, the mean squared error values are 145.73 and 3417.2, respectively, while the R^2 values are 0.95 and 0.97. The R^2 score shows the model's ability to capture data variability. As a result, the high value of R^2 suggests that the created NN model is relatively well suited to the gold price data set. The proximity of R^2 values for both the training and test data sets indicates that the generated model is a generic model that can perform equally well with unknown data. Statistical similarities were also measured to do a statistical comparison between the actual values and the model projected values. As a result, a paired-sample *t*-test was run with the null hypothesis that the means of the two populations differed. Table 2 displays the *t*-statistics results with the null hypothesis' level of significance for the training and test data sets. The significance levels in the table are not less than 0.05, as can be seen. Since there is no observable difference between the means of the training and test sets of gold price data, the null hypothesis may be ruled out. Figure 7 displays the scatter plots of the actual values for the gold price test data set in comparison to the model's projected values. The fact that all of the points in Figure 7 are quite near the bisector line suggests that the predicted values and the actual values are fairly similar. By calculating the auto-correlation function of the gold price test data set's actual and predicted values, the reproduction of auto-correlation was also explored. For both the actual and predicted test data sets, interesting that the auto-correlation is increased after lag 35 in Figure 8. As seen in the figure, the auto-correlation values of the projected data and the auto-correlation values of the test data match up quite well.

Table 1. Statistics of our suggested model's training and testing data errors for predicting the price of gold.

	Training	**Testing**
Mean error	0.86	−2.06
Mean absolute error	8.59	44.59
Mean squared error	145.73	3417.2
Error variance	145.59	3477.3
R^2	0.95	0.97

Table 2. Comparing actual values from the training and test data sets of the proposed model to those expected values in a paired sample *t*-test.

	Training Data	**Testing Data**
Mean	0.86	−2.06
Std. deviation	12.07	58.97
Std. error of the mean	0.77	8.02
95% confidence of the difference	Lower = −0.66, upper = 2.38	Lower = −18.15, upper = 14.03
t-statistic value (*t*)	1.11	−0.26
Degree of freedom (df)	243	53
Significant level (2-tailed)	0.268	0.798

To illustrate the effectiveness of the suggested method compared to existing methods, comparison research was conducted. We chose MLP trained using the gradient descent methodology as a baseline for comparison reasons. We chose both the logistic and the gaussian activation functions for the MLP models. The support vector machine (SVM) algorithm was another technique chosen for comparison. We used a grid search approach to choose the SVM model's hyper-parameters. For comparison, the linear time series forecasting AR technique was also chosen. The 5-fold cross-validation approach was used to train all methods using the training data set, and the test data set was used for testing. All methods' error statistics were determined and are shown in Table 3. In terms of error minimization, our proposed strategy outperformed all other tested methods, according to the table. The SVM model was also discovered to be the second-best model for this investigation, outperforming the MLP and AR models. Performance-wise, the MLP model outperformed the AR model. This may be because the linear AR model failed to account for some nonlinearity in the gold price data. When choosing all of the parameters for non-linear modeling, including the NN model, together using an evolutionary algorithm, the outcomes are better than with traditional non-linear training since the correct selection of many parameters is essential.

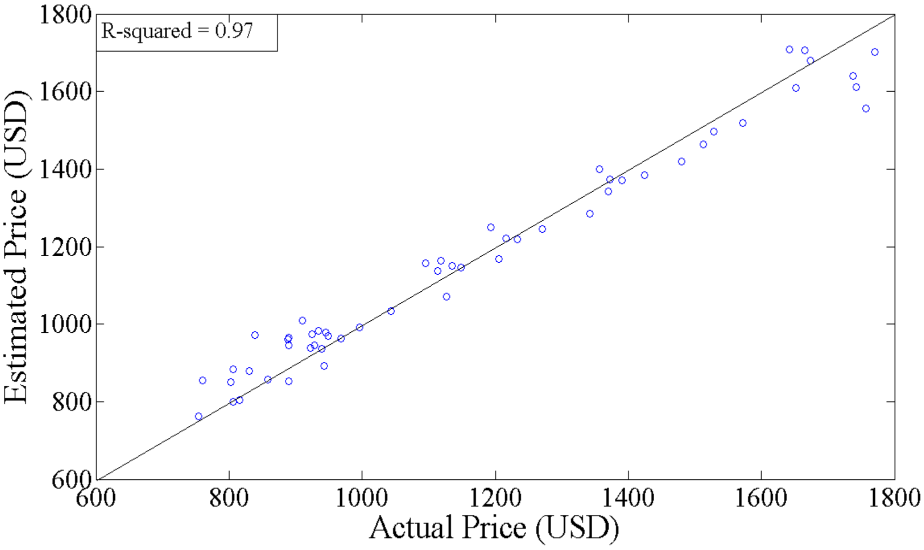


Figure 7. The gold price test data sets observed vs. predicted values are displayed in a scatter plot using an evolutionary-based NN model.

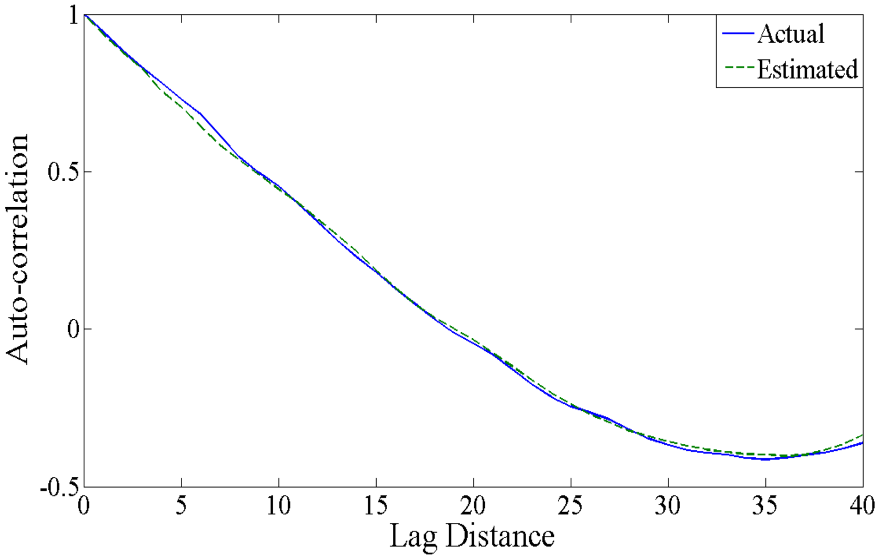


Figure 8. Actual and projected gold price auto-correlation charts for the training data set.

Table 3. Forecasting error statistics with our suggested approach and various other neural network architectures, the AR model, and the SVM model.

Number	MLP (Logistic)	MLP (Gaussian)	Support Vector Machine	AR	Our Proposed Method
Mean error	37.0705	28.6028	22.3169	41.1913	**−2.06**
Mean absolute error	53.4765	48.0972	47.6286	62.3700	**44.59**
Mean squared error	4.9146×10^3	4.1406×10^3	3.8858×10^3	6.2194×10^3	$\mathbf{3.4172 \times 10^3}$
Error variance	3.9268×10^3	3.8610×10^3	3.5087×10^3	4.8457×10^3	$\mathbf{3.4773 \times 10^3}$
$\mathbf{R^2}$	0.8787	0.9264	0.9388	0.7548	**0.97**

The comparison investigation showed that the evolutionary-based algorithm was more effective at predicting metal prices than several common techniques. However, the quantity of input variables used for the NN model has a significant impact on how well the proposed strategy works. An inaccurate forecasting model may be the result of improper input variable selection. In this study, we chose the input variables by the somewhat arbitrary process of determining the value of p using the average entropy value. To confirm our choice, we developed many neural network models using the optimum parameters (H_n = 9, η = 0.026, and μ = 0.76) and varying the number of input variables. Changes in the amount of input variables will also affect the number of pattern sets produced from historical gold price data. The last 100 patterns from the pattern sets were chosen for comparison to be effective. For verification, the fitness function of Equation (9) was applied. The test data set's fitness values are shown in Figure 9 for various combinations of input variables. The graph demonstrates that there are seven input variables, at which point the error value is lowest. It should be noted that when choosing the p-value using average entropy, we also chose the input variables to be 7. This study provides evidence that the right neural network design may be chosen when input variables are chosen using an entropy-based methodology.

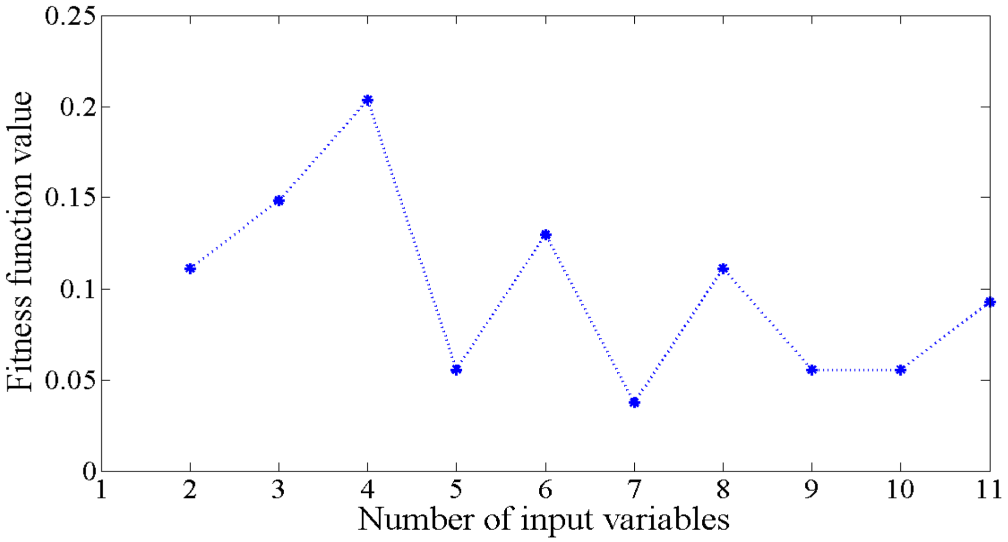


Figure 9. Test data set error values for various input numbers.

To check the optimality of the selected hidden node size (H_n), we developed multiple NN models by varying the hidden node number while fixing other parameters (number of input variables = 7, η = 0.026, and μ = 0.76). We changed the hidden node number from 1 to 25 and calculated the fitness function value of the test data. The fitness function values for various hidden node numbers are shown in Figure 10. The image illustrates that the hidden node number of 9, which is also the value chosen by our suggested evolutionary-based algorithm, results in the lowest fitness function value. This study authenticated the selection of hidden node size using an evolutionary-based algorithm.

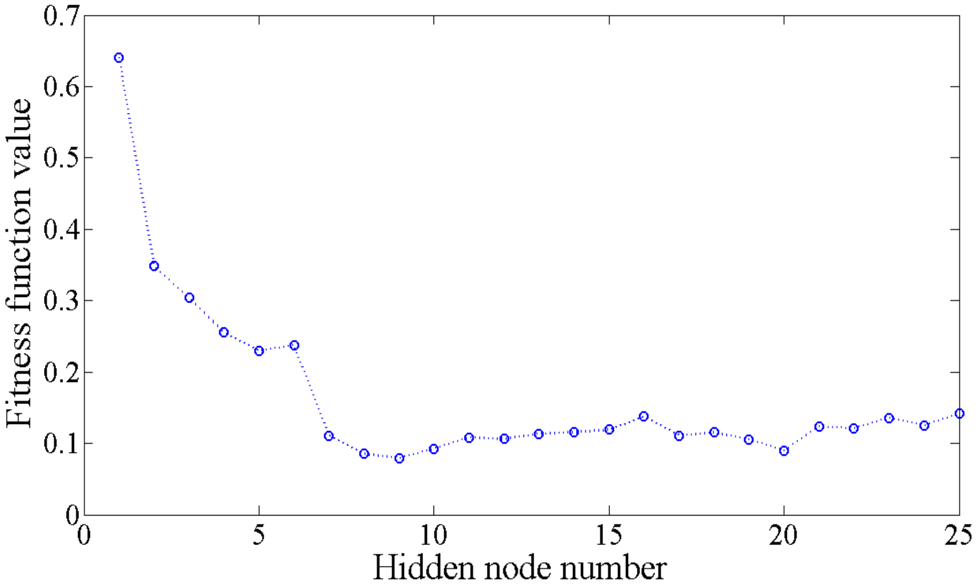


Figure 10. Variations in the 5-fold cross-validation data's error levels due to the number of hidden nodes.

6. Conclusions

In this research, we suggested an NN model for metal price forecasting based on evolutionary algorithms. However, the performance of the non-linear NN models is greatly influenced by a variety of characteristics, including the number of input variables in the forecasting model, the number of hidden nodes, the learning rate, and momentum, among others. The results of implementing the entropy-based method demonstrate that this strategy can aid in selecting the input parameters for NN-based metal price forecasting models in the most effective manner. The evolutionary algorithm aids in choosing the parameters and the NN model's hidden node design. This research shows that by avoiding the trial-and-error process of parameter selection, the best-chosen network design and network parameters considerably save computational time while improving the performance of the model. The evolutionary-based NN model was created to predict the price of gold. The outcomes showed that the technique can be successfully used to project the price of gold in the future. According to the study's findings, an evolutionary-based NN model with carefully chosen network architecture, input variable numbers, and NN model parameters can greatly boost the model's performance. The benefit of the suggested method is that, unlike linear time series forecasting methods, no a priori assumption about the distribution of input and output variables is necessary. A comparative study was conducted to assess the prediction abilities of several linear and non-linear time series models. The comparative study's findings show that our suggested strategy outperformed the other metal price forecasting techniques. The findings of this study have proved the effectiveness of this strategy in predicting gold price forecasts. The drawback is that the evolutionary method used in this research assumes constant values for the crossover rate and mutation rate, among other parameters. However, these variables also contribute to the creation of the model. The performance of the model might be enhanced by careful parameter selection.

Author Contributions: Conceptualization, D.J., P.C. and D.A.; methodology, D.J., P.C. and K.A.; software, D.J., P.C. and F.H.; validation, D.J., P.C. and V.Y.T.; analysis, D.J., P.C. and K.A.; investigation, E.B.T.; resource, F.H. and V.Y.T.; writing—original draft preparation, D.J. and P.C.; writing—review and editing, D.J., P.C., E.B.T. and F.H.; supervision, P.C. and D.A.; project administration, V.Y.T. and E.B.T. All authors have read and agreed to the published version of the manuscript.

Funding: Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2023R236), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Data Availability Statement: The data utilized in this study is collected from https://www.indexmundi.com (accessed on 18 December 2022) and authors would like to thank the support of NIT Rourkela for providing the environment to carry out the case study of this paper.

Acknowledgments: Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2023R236), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chengbao, W.; Yanhui, C.; Lihong, L. The Forecast of Gold Price Based on the GM (1, 1) and Markov Chain. In Proceedings of the IEEE International Conference on Grey Systems and Intelligent Services, Nanjing, China, 18–20 November 2007; pp. 739–743.
2. Roberts, M.C. Duration and characteristics of metal price cycles. *Res. Pol.* **2009**, *34*, 87–102. [CrossRef]
3. Xiao, F.; Jin, W. Mineral Price Forecasting Based on Nonlinear Regression. *Nonferrous Met. Sci. Eng.* **2011**, *2*, 72–75.
4. John, E.T. Long-term trends in copper prices. *Mining Eng.* **2002**, *54*, 25–32.
5. Shahriar, S.; Erkan, T. An overview of the global gold market and gold price forecasting. *Res. Pol.* **2010**, *35*, 178–189.
6. Chen, L.; Zhang, X. Gold Price Forecasting Based on Projection Pursuit and Neural Network. *J. Phys. Conf. Ser.* **2019**, *1168*, 062009. [CrossRef]
7. Mombeini, H.; Yazdani-Chamzini, A. Modeling gold price via artificial neural network. *J. Econ. Bus. Manag.* **2015**, *3*, 699–703. [CrossRef]
8. Yang, J.; Montigny, D.D.; Treleaven, P.C. ANN, LSTM, and SVR for Gold Price Forecasting. In Proceedings of the IEEE Symposium on Computational Intelligence for Financial Engineering and Economics (CIFEr), Helsinki, Finland, 4–5 May 2022; pp. 1–7. [CrossRef]
9. Box, G.; Jenkins, G. *Time Series Analysis: Forecasting and Control*; Holden-Day: San Francisco, CA, USA, 1970.
10. Dooler, G.; Lenihan, H. An assessment of time series methods in metal price forecasting. *Res. Pol.* **2005**, *30*, 208–217. [CrossRef]
11. Jianhong, C.; Xueyan, Y.; Shan, Y.; Lang, L. Analysis and Forecast About Mineral Product Price Based on Time Series Model. *Kunming Univ. Sci. Technol.* **2009**, *34*, 9–14.
12. Biao, Y.; Zhouquan, L.; Guang, L.; Yiwei, Y. Open pit mining limit dynamic optimization based on economic time series forecasting. *China Coal Soc.* **2011**, *36*, 29–33.
13. Mcdonald, J.; Xu, Y. Some forecasting applications of partially adaptive estimators of ARIMA models. *Econ. Lett.* **1994**, *45*, 155–160. [CrossRef]
14. Chunmei, L. Price Forecast for Gold Futures Based on GA-BP Neural Network. In Proceedings of the IEEE International Conference on Management and Service Science, Beijing, China, 20–22 September 2009.
15. Lian, Z.; Dandi, M.; Zongxin, L. Gold Price Forecasting Based on Improved BP Neural Network. *Comput. Simul.* **2010**, *27*, 200–203.
16. Masters, T. *Advanced Algorithms for Neural Networks*; John Wiley and Sons: New York, NY, USA, 1995.
17. Shijiao, Y.; Changzhen, W.; Jianyong, D.; Xiaoyu, H. Comparative Analysis of Price Prediction of Molybdenum Products Based on ANFIS and BPNN. In Proceedings of the IEEE International Conference on Computational Intelligence and Industrial Application, Nanning, China, 11–14 December 2010; pp. 126–129.
18. Cortez, P.; Rio, M.; Rocha, M.; Sousa, P. Internet traffic forecasting using neural networks. In Proceedings of the 2006 International Joint Conference on Neural Networks (IJCNN 2006), Vancouver, BC, Canada, 16–21 July 2006; pp. 4942–4949.
19. Hassan, M.R.; Nath, B.; Kirley, M. A fusion model of HMM, ANN, and GA for stock market forecasting. *Exp. Syst. Appl.* **2007**, *33*, 171–180. [CrossRef]
20. Haykin, S. *Neural Networks: A Comprehensive Foundation*; Prentice Hall: Hoboken, NJ, USA, 1999.
21. Niska, H.; Hiltunen, T.; Karppinen, A.; Ruuskanen, J.; Kolehmainen, M. Evolving the neural network model for forecasting air pollution time series. *Engg. Appl. Art. Intell.* **2004**, *17*, 159–167. [CrossRef]
22. Jain, A.; Kumar, A.M. Hybrid neural network models for hydrologic time series forecasting. *Appl. Soft Comput.* **2007**, *7*, 585–592. [CrossRef]
23. Livieris, I.E.; Pintelas, E.; Pintelas, P. A CNN–LSTM model for gold price time-series forecasting. *Neural Comput. Appl.* **2020**, *32*, 17351–17360. [CrossRef]
24. Guha, B.; Bandyopadhyay, G. Gold price forecasting using ARIMA model. *J. Adv. Manag. Sci.* **2016**, *4*, 117–121.
25. Alameer, Z.; Elaziz, M.E.; Ewees, A.A.; Ye, H.; Jianhua, Z. Forecasting gold price fluctuations using improved multilayer perceptron neural network and whale optimization algorithm. *Res. Pol.* **2019**, *61*, 250–260. [CrossRef]
26. Chai, J.; Zhao, C.; Hu, Y.; Zhang, Z.G. Structural analysis and forecast of gold price returns. *J. Manag. Sci.* **2021**, *6*, 135–145. [CrossRef]
27. Das, S.; Sahu, T.P.; Janghel, R.R. Oil and gold price prediction using optimized fuzzy inference system based extreme learning machine. *Res. Pol.* **2002**, *79*, 103109. [CrossRef]
28. Wedding, D.K.; Cios, K.J. Time series forecasting by combining networks, certainty factors, RBF and the Box–Jenkins model. *Neurocomputing* **1996**, *10*, 149–168. [CrossRef]
29. Armano, G.; Marchesi, M.; Murru, A. A hybrid genetic-neural architecture for stock indexes forecasting. *Inf. Sci.* **2005**, *170*, 3–33. [CrossRef]

30. Pelikan, E.; DeGroot, C.; Wurtz, D. Power consumption in West-Bohemia: Improved forecasts with decorrelating connectionist networks. *Neural Netw. World* **1992**, *2*, 701–712.
31. Ginzburg, I.; Horn, D. Combined neural networks for time series analysis. *Adv. Neural Inf. Prog. Syst.* **1994**, *6*, 224–231.
32. Goh, W.Y.; Lim, C.P.; Peh, K.K. Predicting drug dissolution profiles with an ensemble of boosted neural networks: A time series approach. *IEEE Trans. Neural Netw.* **2003**, *14*, 59–463.
33. Yu, L.; Wang, S.; Lai, K.K. A novel nonlinear ensemble forecasting model incorporating GLAR and ANN for foreign exchange rates. *Comp. Oper. Res.* **2005**, *32*, 2523–2541. [CrossRef]
34. Crone, S.F.; Kourentzes, N. Feature selection for time series prediction—A combined filter and wrapper approach for neural networks. *Neurocomputing* **2010**, *73*, 1923–1936. [CrossRef]
35. Zhang, G.; Patuwo, B.E.; Hu, M.Y. Forecasting with artificial neural networks: The state of the art. *Int. J. For.* **1998**, *14*, 35–62.
36. DeGroot, C.; Wurtz, D. Analysis of univariate time series with connectionist nets: A case study of two classical examples. *Neurocomputing* **1991**, *3*, 177–192. [CrossRef]
37. Bishop, M. *Neural Networks for Pattern Recognition*; Clarendon Press: Oxford, UK, 1998.
38. Cortez, P.; Rocha, M.; Neves, J. Evolving Time Series Forecasting ARMA Models. *J. Heuristics* **2004**, *10*, 415–429. [CrossRef]
39. Chen, Y.; Chang, F.J. Evolutionary artificial neural networks for hydrological systems forecasting. *J. Hydro* **2009**, *367*, 125–137. [CrossRef]
40. Panagiotopoulos, D.; Orovas, C.; Syndoukas, D. Neural Network Based Autonomous Control of a Speech Synthesis System. *Intell. Sys. App.* **2022**, *14*, 200077. [CrossRef]
41. Koike, K.; Matsuda, S.; Gu, B. Evaluation of Interpolation Accuracy of Neural Kriging with Application to Temperature-Distribution Analysis. *Math. Geo.* **2001**, *33*, 421–448. [CrossRef]
42. Xue, Y.; Tong, Y.; Neri, F. An ensemble of differential evolution and Adam for training feed-forward neural networks. *Inf. Sci.* **2022**, *608*, 453–471. [CrossRef]
43. Xue, Y.; Wang, Y.; Liang, J.; Słowik, A. A Self-Adaptive Mutation Neural Architecture Search Algorithm Based on Blocks. *IEEE Comp. Intell. Mag.* **2021**, *16*, 67–78. [CrossRef]
44. Xue, Y.; Qin, J. Partial Connection Based on Channel Attention for Differentiable Neural Architecture Search. *arXiv* **2022**, arXiv:abs/2208.00791. [CrossRef]
45. Abdolrasol, M.G.M.; Hussain, S.M.S.; Ustun, T.S.; Sarker, M.R.; Hannan, M.A.; Mohamed, R.; Ali, J.A.; Mekhilef, S.; Milad, A. Artificial Neural Networks Based Optimization Techniques: A Review. *Electronics* **2021**, *10*, 2689. [CrossRef]
46. MacKay, D.J.C. *Information Theory, Inference, and Learning Algorithms*; Cambridge University Press: Cambridge, UK, 2003.
47. Chatterjee, S.; Bandopadhyay, S. Reliability estimation using a genetic algorithm-based artificial neural network: An application to a load-haul-dump machine. *Exp. Sys. App.* **2012**, *9*, 10943–10951. [CrossRef]
48. Abdullah, L. ARIMA Model for Gold Bullion Coin Selling Prices Forecasting. *Int. J. Adv. Appl. Sci.* **2012**, *1*, 153–158. [CrossRef]
49. Khan, M.M. Forecasting of Gold Prices (Box Jenkins Approach). *Int. J. Emer. Tech. Adv. Eng.* **2013**, *3*, 662–670.
50. Davis, R.; Dedu, V.K.; Bonye, F. Modeling and Forecasting of Gold Prices on Financial Markets. *Am. Int. J. Contemp. Res.* **2014**, *4*, 107–113.
51. Tripathy, N. Forecasting Gold Price with Auto Regressive Integrated Moving Average Model. *Int. J. Eco. Fin. Issues* **2017**, *7*, 324–329.
52. Šimáková, J. Analysis of the Relationship between Oil and Gold Prices. *J. Fin.* **2011**, *51*, 651–662.
53. Apergis, N. Can gold prices forecast the Australian dollar movements? *Int. Rev. Econ. Fin.* **2014**, *29*, 75–82. [CrossRef]
54. Cai, J.; Cheung, Y.L.; Wong, M.C. What Moves the Gold Market. *J. Futures Mark.* **2001**, *21*, 257–278. [CrossRef]
55. Levin, E.J.; Wright, R.E. *Short-Run and Long-Run Determinants of the Price of Gold*; Research Study No. 32; World Gold Council: London, UK, 2006.
56. Baker, S.A.; Van Tassel, R.C. Forecasting the Price of Gold: A Fundamentalist Approach. *Atl. Eco. J.* **1985**, *13*, 43–51. [CrossRef]
57. Lawrence, C. *Why is Gold Different from Other Assets? An Empirical Investigation*; World Gold Council: London, UK, 2003.
58. Tully, E.; Lucey, B.M. A power GARCH examination of the gold market. *Res. Int. Bus. Fin.* **2007**, *21*, 316–325. [CrossRef]
59. Parisi, A.; Parisi, F.; Diaz, D. Forecasting old price changes: Rolling and recursive neural network models. *J. Multi. Fin Manage.* **2008**, *18*, 477–487.
60. Honarkha, M.; Caers, J. Stochastic simulation of patterns using distance-based pattern modeling. *Math. Geosci.* **2010**, *42*, 487–517. [CrossRef]
61. Holland, J. *Adaptation in Natural and Artificial Systems*; The University of Michigan Press: Ann Arbor, MI, USA, 1975.
62. Dengiz, B.; Atiparmak, F.; Smith, A.E. Local search genetic algorithm for optimization of highly reliable communications networks. *IEEE Trans Evol. Comp.* **1997**, *1*, 179–188. [CrossRef]
63. Li, J.P.; Balazs, M.E.; Parks, G.T. Engineering design optimization using species-conserving genetic algorithms. *Eng. Optm.* **2007**, *39*, 147–161. [CrossRef]
64. Congdon, C.B. A comparison of Genetic Algorithm and Other Machine Learning Systems on a Complex Classification Task from Common Disease Research. Ph.D. Thesis, University of Michigan, Ann Arbor, MI, USA, 1995.

65. Willighagen, E. R Based Genetic Algorithm for Binary and Floating-Point Chromosomes. 2005. Available online: http://rss.acs.unt.edu/Rdoc/library/genalg/html/00Index.html (accessed on 18 December 2022).
66. Limas, M.C. A MORE Flexible Neural Network Package. 2006. Available online: http://wiki.rproject.org/rwiki/doku.php?id=packages:cran:amore (accessed on 18 December 2022).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.