Journal of Scientific & Industrial Research
Vol. 83, April 2024, pp. 432-444
DOI: 10.56042/jsir.v83i4.5559

A Hybrid Model of Neural Networks and Genetic Algorithms for Prediction of the Stock Market Price: Case Study of Palestine

Lama AlQasrawi[1], Mohammed Awad[2]* & Rami Hodrob[2]

[1]Department of Computer Science, Arab American University, Palestine

[2]Department of Computer Systems Engineering, Arab American University, Palestine

Received 14 September 2023; revised 15 February 2024; accepted 11 March 2024

Accurate stock market predictions are critical to investor protection and economic growth. This study is the first of its kind to anticipate Palestinian stock market values using artificial intelligence models. In this paper, an improved hybrid model is given that combines multilayer perceptron neural networks with genetic algorithms to predict the state of the Palestinian stock market using the Al-Quds Index as the major indicator (MLPNNs-GAs). Furthermore, the stock values of the three largest Palestinian companies will be forecast using their stock market data. The rationale for merging artificial neural networks (ANNs) and genetic algorithms (GAs) stem from the fact that stock price data bear highly volatile and nonlinear features. The undiscovered patterns of relationships in the input and output data can be explored by artificial neural networks. The weights for the NNs are optimized using genetic algorithms (GAs), which determine the optimal weights based on performance and best-predicted minimal mean square error (MSE) value. Recurrent neural networks with Levenberg-Marquardt (RNNs-LM) and MLPNNs-LM, two more classic models of various neural network techniques, were used to compare the prediction performance of the proposed model in terms of mean square error. The experimental results show that, with MSEs of 0.0011 for the Al-Quds Index, 0.0021 for the Bank of Palestine, 0.001 for Palte, and 0.0006 for Padico, the recommended hybrid model MLPNNs-GAs outperforms other models in terms of closing price predictions. It has been shown that the MLPNNs-GAs model may give stock market investors reliable and accurate tools for making forecasts; as a result, MLPNNs-GAs is advised as an effective model for the prediction of nonlinear financial time series data.

Keywords: Forecasting, Genetic algorithms, Levenberg marquardt algorithm, Multilayer perceptron NNs, Recurrent NNs

Introduction

The stock markets are vital hubs for developing nations, and successful stock market management requires a shift from saving to investing through share purchases and sales. These activities are so important that they are impacted by social, political, cultural, and economic considerations. Therefore, it is reasonable to consider the value of an action in an instant not as a deterministic variable but as a random variable, considering its time trajectory as a stochastic process.[1] Researchers and investors find it challenging to predict the direction of the stock market index due to the expansion of societies, the increasing complexity of the economy, and the enduring presence of conflict and instability.[2] The nation's stock market is typically examined through the lens of advanced economics, computer science, and mathematical understanding. In particular, predicting the movements of the stock market and financial market has become essential to assist investors in making wise choices during their future planning, which eventually opens up new avenues for profit and safeguards against possible losses.[3] Time series, or sets of observations arranged according to the sequence in which they occur, serve as the foundation for this forecasting. Thus, in order to find patterns that can be utilized to set expectations for the future and guide pertinent decision-making, causes and consequences are analyzed. Regarding the theoretical perspective of the modeling and forecasting of this type of financial time series, there are two common positions. The first assumes that enough relevant information about a financial asset's price makes it more predictable. Also, the historical performance of the assets will likely recur in the future. The Efficiency Market Hypothesis, which contends that previous prices cannot be used to forecast future prices, is the second. It guarantees that markets accurately reflect all available information and are unable to adhere to a trade regulation that offers further advantages. Accepting the efficiency market hypothesis would imply that it is impossible to exceed the market average. However, in practice, this was

*Author for Correspondence
E-mail: mohammed.awad@aaup.edu

refuted by investors like Warren Buffet and Peter Lynch, who managed to exceed the market average for more than 20 years.[4]

The two methods for analyzing stock markets are technical analysis and fundamental analysis. The first uses numerical analysis based on time series to project the prices that equities with historical returns will achieve. A study of the variables influencing supply and demand constitutes the second. It is accomplished by analyzing, gathering, and interpreting the data that businesses provide in the form of news, balance sheets, reports, and statements. To date, the literature in this area has focused on these two strategies independently, employing computational, statistical, or econometric methods to produce short-, medium-, or long-term forecasts.[5,6]

A prediction model has been put forth by a number of scholars to forecast stock exchange trends across several foreign stock markets. While some rely on their research on single models such as neural networks, genetic algorithms, particle swarm optimization algorithms, and neuro-fuzzy systems, others rely on hybrid approaches that combine two or more Artificial Intelligence (AI) techniques.[7] The basic idea of the hybrid model is to overcome the weaknesses of the single models, take advantage of each model, and generate more accurate results.

The researcher's ultimate objective is to predict the financial time series corresponding to the Palestinian market through hybrid models of intelligent systems. Two intelligent methods, MLPNNs and GAs, are combined to improve the forecasting accuracy of the stock exchange in Palestine. Better forecasting of the Palestinian stock market in terms of Mean Square Error (MSE) values will be feasible thanks to the unique proposed hybrid model "MLPNNs-GAs," which adjusts the GAs phases to determine the MLPNN's weight. Additionally, the suggested hybrid model performs better than the two applied models (MLPNNs-LM and RNNs-LM), which helps identify the most effective model for predicting the price of stocks on the Palestinian Stock Market. On the other hand, the datasets were collected from the Palestinian stock market. In order to identify patterns in these markets and forecast the price series of future stock markets, the price-time series of the shares in these stock markets will be analyzed based on historical data exchange intervals. The procedure begins with arranging the neural network architecture, followed by the hybrid system, training the suggested models, projecting future values, and evaluating the suggested model's effectiveness in comparison to other ANN models (MLPNNs-LM model and RNNs-LM model).

Related Works

This paper will discuss an important problem regarding the research topic. The problem is investigating the method used for economic condition forecasting. In time series forecasting, the main problem is the use of effective methods that fit with the time series, depending on the nature of the data. All financial data is non-linear, and the use of traditional statistical methods such as regression analysis and ARIMA models may generate problems. These techniques may produce data that is erroneous; therefore, projecting the future state of the economy may not be as appropriate as previously believed.[8] Thus, it is important to consider the necessary and most efficient methods when the data is non-linear. Makridakis *et al*. stated that the complex problem is solved by using artificially intelligent models with no need for logical, traditional, or statistical methods.[9]

During the last few years, many researchers have tried to predict the stock markets in different countries using intelligent methods. Many researchers depend on their studies of datasets collected from international stock exchange markets. While few researchers depend on the Palestinian stock exchange market in their studies, Tay *et al.*[10] investigated the application of SVM in financial forecasting compared with ARIMA and ANN. The data set was collected from the Al-Quds Index of the Palestinian Stock Exchange Market to forecast two-month future points. They concluded that SVM produces a better forecasting result than the other two models. Lahmiri *et al*. compared the accuracy of three hybrid intelligent systems in forecasting ten international stock markets. They used genetic algorithms with Adaptive Time-delay Neural Networks (ATNN), Time-Delay Neural Networks (TDNN), and the Adaptive Neuro-Fuzzy Inference System (ANFIS). The results showed that the ANFIS model produced the most accurate forecasts for the future price of seven international stock markets out of the ten. GA-TDNN usually produces better forecasting results compared with GA-ATDN.[11] Boyacioglu *et al*. investigated the predictability of Istanbul stock market returns with the Adaptive-Network-Based Fuzzy Inference System. The results reveal that the model successfully forecasts the monthly returns of the ISE National 100 Index with an accuracy rate of 98.3%.[12]

Hu *et al*. performed a study to compare the forecast's accuracy in trend analysis between the forecast using Google Trends data and the forecast without Google. To estimate the stock price trend for the S&P 500 and DJIA Indices, they created a hybrid forecasting method called the ISCA-BPNN model. They did this by combining the Sine Cosine Algorithm (ISCA) with BPNN to improve the weights and basis for the BP network. The collected results demonstrated this hybrid model's appropriateness and indicated that stock prediction is improved by Google Trends.[13] Aamodt *et al*. compared the performance of FFNN, ESN, CRBM, TDNN, and CNN. SVM and these techniques are also contrasted, when used to project the price of stocks for a brief period of time. Stock data is sourced from markets that are increasing or falling and listed on (NYSE) or (NASDAQ). According to their findings, FFNN, CNN, and SVM produce better outcomes and profit. Low revenues where produced from alternative techniques.[14] Chhajer *et al*.[15] conducted a study to discussed the strengths and weaknesses of ML for stock market prediction and provided some insight into the opportunities and threats in applying advanced technologies for stock market prediction, studied the applications of NNs, support vector machines, and long–short-term memory in stock market prediction.

Pradeepkumar *et al*.[16] offered a Quantile Regression Neural Network (QRNN) model trained on Particle Swarm Optimization (PSO) to forecast volatility and instability in financial time series. The other seven volatility-predicting models—GARCH, MLP, GRNN, GMDH, RF, QRRF, and QRNN were compared to this model's performance. Eight financial datasets provided the information needed to evaluate the performance of these suggested models. The outcomes demonstrated that PSOQRNN performed better than these models in terms of (MSE). Nayak *et al*.[17] developed an adaptive single-layer second-order neural network with GAs-based training (ASON-GAs) and compared performance with another two developed methods ANN, and RNN. They demonstrate that the proposed model does better forecasting accuracy in handling uncertainties and nonlinearities. Ouyang *et al*.[18] suggested an SVM based on uncertain knowledge that was optimized via particle swarm optimization (PSO-UK), and evaluated the PSO-UK-SVM and PSO-SVM prediction accuracy in forecasting the SSE composite index. The results indicate that the first mentioned model produced the results of the most accurate forecasts. In Wang *et al*.[19] the authors depend on the Back Propagation neural network to propose a forecast model of the Shanghai composite index. This model has higher accuracy strong robustness and a higher capacity of fault tolerance. They found that the model was good at forecasting. A hybrid intelligent system has been proposed by Atalakis *et al*.[20] based on ANNs and a neuro-fuzzy system for analyzing the trend prediction of the market. The results showed the efficiency of this hybrid intelligent system.

A few academics concentrated on predicting the Palestine stock market. After reviewing this specific study area, it is discovered that not many studies had employed a hybrid model that combines evolutionary algorithms and neural networks as a prediction tool. Consequently, a more effective model that aids in predicting the non-linear data will help raise the accuracy of forecasts for the Palestinian stock market and aid in decision-making.

Methodology

The study used daily and weekly closing price data for four datasets from the Palestinian Stock Exchange, excluding holidays. The study included the daily closing price data for three of the biggest companies on the stock exchange. These companies were selected among the most active and traded companies in the Palestine Exchange (the period from 2010–2017 for both Padico and Bank of Palestine companies and from 2011–2017 for Paltel company). The weekly closing price data for the main index of the Palestinian Stock Exchange is the Al-Quds Index (a period from 2010–2017). The overall protocol followed to conduct the study is shown in Fig. 1. Together with the time series of the Al-Quds Index, which shows the behavior of the companies in the Palestine stock market, the input data will be used to feed the model. This data presents a time series of the stock market behavior of these companies. The data will next go through a pre-processing step, where the issue of missing data for each dataset is first addressed. This is done by substituting the average value of the data for the missing data on the same day as the previous application year. Secondly, normalization was done to adapt the data to the output range of the logistic activation function. It was scaled in the range of 0–1 using the following general formula[21]:

$$d_i = \frac{(d - \min(d))}{(\max(d) - \min(d))} \quad \ldots (1)$$

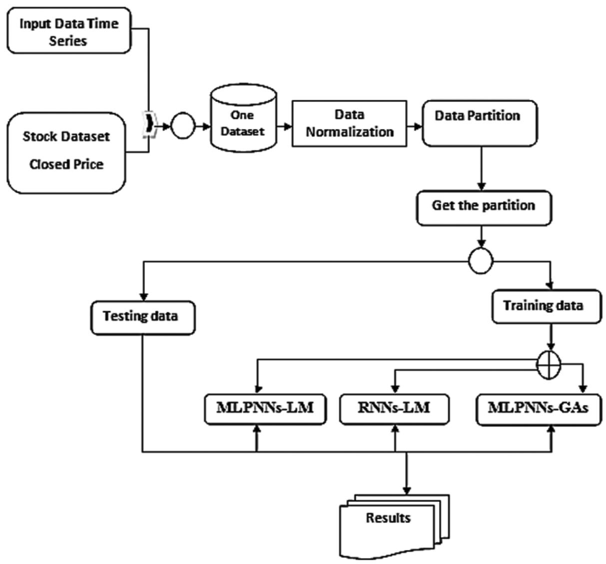

Fig. 1 — General procedure used in conducting experiments

where, d is the original value and, d-i is the normalized value, max(d) and, min(d) are the maximum and minimum values of the original dataset. After pre-processing datasets, the data was split into two parts using cross-validation. The initial one used 70% of the total data's input-target pairs for training, whereas the second one used 30% (testing). Since each firm has a varied period (number of years and days in a year), the amount of input data for testing and training in each company is different from the others. The three applicable models' performance evaluations will be verified using data from testing and training.

This work will improve the prediction process of the stock market price. The mean square error (MSE) was used for evaluating the model performance using Eq. 2.

$$MSE = \frac{1}{m}\sum_{a=1}^{m}\sum(Ta_a - Op_a)^2 \quad \ldots(2)$$

where, m represents the total number of input data, Ta_a is the actual value, and Op_a is the predicted value.

Applied Models

Several factors affect future investments and have high effects on market movements up or down, which include wars, natural disasters, conflicts, anxiety over inflation or deflation, and changes and developments in technology. Investors also want techniques for future prediction because the investment process is unpredictable and dangerous. This will help them make well-informed decisions, such as whether to enter or depart the market, with a high degree of confidence. The creation of an estimated and outcome-based set of future conditions and events which could be favorable or unfavorable that can be considered when making plans is the foundation of the financial forecast. The estimates are developed based on scientific and statistical methods. Using historical data, estimates help in determining the changes in the environment surrounding investors and thereby assist them in making investment decisions to reach profits with minimum risks. ANN models have the ability to process data without the need for a certain structure or model. A description of ANN prediction techniques such as MLPNNs-LM, RNNs-LM, and hybrid model MLPNNs-GAs is introduced in the following sections.

Multilayer Perceptron Neural Networks-LM (MLPNNs-LM)

There are two stages to this model: feed-forward bass and backward bass. The forward pass includes the processes to compute the results of the input layer, the layer that comes after, and the outcome layer. The actual and intended result is not convergent during this phase since random weights are employed. By adjusting the weights, backward bass reduced the performance function's result by making the discrepancy between the actual and intended target (gradient decent error) sufficiently small[22].The general structure of the MLPNNs illustrated in Fig. 2. The General formulas for computing outcome of the first hidden layer y_1 of the MLPNNs are:

$$\mathrm{outc}_{y1} = \sigma^1(\textstyle\sum_{i=1}^{m} X_i . K_{iy1}) \quad \ldots(3)$$

where, m represents the total number of input data, xi represents the input vector and k represents the weight vectors. The general formula for computing the outcome of the final outcome layer $y2$ is illustrated in Eq. 4, here $outc_{y1}$ is input to the final outcome layer $y2$.

$$\mathrm{outc} = \sigma^2(\textstyle\sum_{j=1}^{m} \mathrm{outc}_{y1} . K_{jy2}) \quad \ldots(4)$$

where, σ^1 and σ^2 are the transfer functions for the hidden and the outcome layers, which computed using these formulas:

$$\sigma^1 = \frac{1}{1+e^{-x}} \quad \ldots(5)$$

$$\sigma^2 = X \quad \ldots(6)$$

The error that occurs for each neuron was calculated as follows:

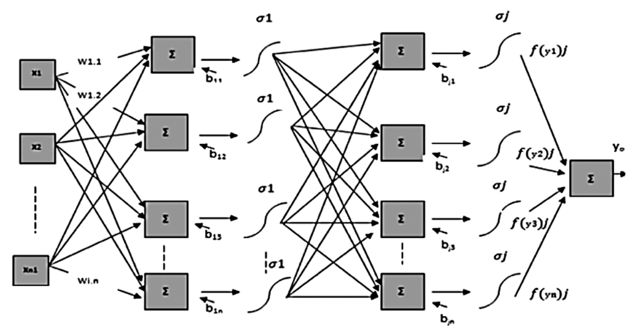

Fig. 2 — General structure of the MLPNNs

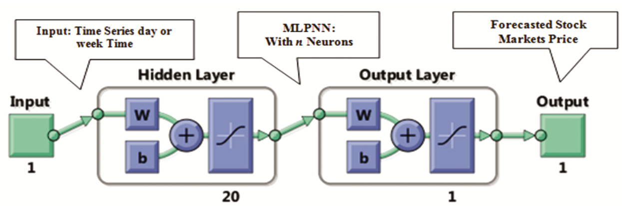

Fig. 3 — Proposed stock markets price using MLPNNs

$$EN_Y = \left(y_k - y_g\right)\sigma^{s'}(Y) \quad \dots (7)$$

where, *EN* is the error between the target outcome and desired outcome, $\sigma^{s'}(Y)$represents the derivative of the transfer function in each layer of the MLPNNs, for the sigmoid $\sigma^{s'}(Y)$ equals $= y_g\left(1 - y_g\right)$. In Eq.8, the process of weights update is illustrated

$$K_Y(t+1) = K_Y(t) + \alpha. EN_Y. y_g \quad \dots (8)$$

where, α represents the learning rate, $K_Y(t)$ represents the value of the current weight, and $K_Y(t+1)$ represents a value of the new weight. For training the MLPNNs, the fastest back propagation method (the Levenberg Marquardt algorithm) was used. The general process of MLPNNs model is represented as shown in Fig. 3.

Levenberg-Marquardt algorithm: In neural network optimization problems, the Levenberg-Marquardt algorithm is an effective training algorithm that determines and adjusts the optimal weights to minimize network error (LMA).[23] A nonlinear least-squares problem is solved using this method. It finds the minimum of a function by blinding both minimization methods, the gradient descent and the Gauss-Newton (GN).[24]

Recurrent Neural Networks-LM (RNNs-LM) Model

Another type of NNs is called recurrent neural networks (RNNs).[25] RNNs are considered as a robust method that is widely used for sequential such as financial data.[26] The concept of this network is to address the shortcomings of existing algorithms; RNN neurons have short-term memory that allows them to retain and utilize the values they acquired from earlier neurons in the future. Thus, the present and the recent past are the two inputs of this algorithm. This is crucial since the sequential data provides valuable insight into what will happen next. Since the predictions in this scenario are dependent upon one another, the model needs to be aware of all prior predictions. Therefore, this network can process a nonlinear time series that depends on prior information and computations to know how to proceed. Data can be processed from the input layer to the output layer thanks to the RNN design. This is comparable to the feed-forward network's direction. Additionally, by having feedback loops, data can be processed in the feed-forward network in the opposite direction, from the outcome layer to the input layer. Thus, the choice made by the h-1 layer influences the neurons in the hidden layer h. As a result, the present input is dependent upon the merging of all recurrent neurons' prior inputs that were previously stored. So, there are correlations between the current, next, and previous data steps in RNNs. Additionally, it employed the Levenberg-Marquardt algorithm to train the RNNs-LM. The general process for RNNs is illustrated in Fig. 4.

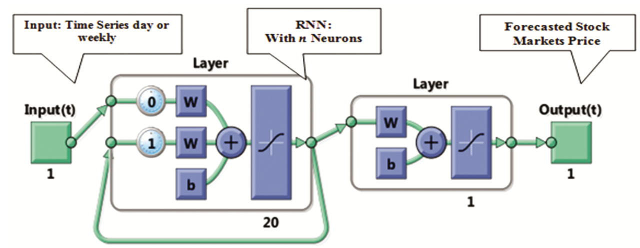

Fig. 4 — Proposed stock markets price using RNNs

The Proposed MLPNNs-GAs Model

Utilize the back-propagation approach to forecast stock exchanges by combining MLPNNs with evolutionary algorithms like GAs. They are highly suitable for the MLPNN training problem, which may be applied in multiple scenarios. In this work, one example is allowing GAs to choose the best weights for the applied MLPNNs. Therefore, the parameters in NNs must be chosen and changed until the best solution with the lowest error value is reached before the training process begins. Because they assess each individual separately, the entities in GAs for this

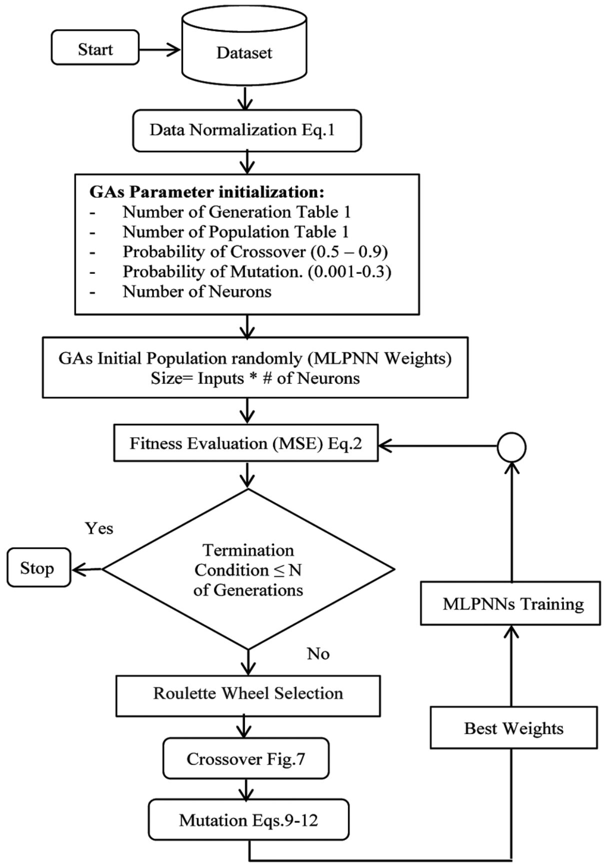

Fig. 5 — MLPNNs-GAs model flowchart

particular challenge arrive at the ideal genome and parameters in NNs. Moreover, GAs is hence suitable for enhancing and developing NNs. Enhancement is achievable by adjusting the number of hidden layers, parameters, and NN topology. To optimize parameters through optimal weights of NNs, it employed the genetic algorithm optimization method in the current work. The MLPNNs-Gas model is presented in Fig. 5. The architecture of the MLPNNs-GAs model includes extracting weight using GAs to use them as initial weights for the MLPNNs-LM.[27] The GAs-weight extraction processes The GAs-weight extraction process as follows:

i. Input the number of generations, the number of populations, the preprocessing dataset, and the number of neurons.
ii. Apply the basic operation of a genetic algorithm.
iii. Output (GAs-weights) to use as initial weights for the neural network.

The basic process for GAs is as follows:

Initialization: In the first step, the genetic algorithms construct the initial population

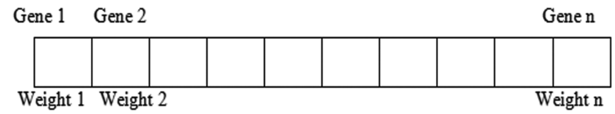

Fig. 6 — Chromosome illustration of the weights

(chromosomes or parents) at random. As shown in Fig. 6, each chromosome has several genes that correspond to the weights of MLPNNs. Gene values are used to indicate each weight. The chromosome must include (x+s) h genes if NNs have x input neurons, h has hidden neurons, and s outcome neurons.

The second phase involves evaluating each chromosome's fitness function in order to assess performance. The mean square error serves as the fitness function (MSE). Applying selection is the third phase; the goal is to select just the most effective chromosomes that match the situation in order to pass on the best traits to the following generation.

Subsequently, in the fourth and fifth processes, crossover and mutation are employed. These involve swapping out the chosen gene and altering the gene in a solution in an attempt to get a better one. Next, compare the results of the effective chromosomes that were chosen in step 3 in a six-step process. Ultimately, iterate through steps 3 through 6 a certain number of times until the ideal answer is achieved. The migration fraction and crossover rate ranges (0.5–0.9) are modified in this work (0.001, 0.002, 0.003, 0.01, 0.02, 0.03, 0.1, 0.2, and 0.3). As found in the earlier research, these values were the best, so the values were chosen via trial and error. Generations and the population have both been altered for this experiment, as shown in Table 1.

In order to choose the optimal groups of these functions with the lowest MSE values, the combination of GAs functions, such as crossover, mutation, and selects, has also been altered as shown in Table 2.

Next, a group of genetic algorithmic functions specified in the MLPNNs-GAs model were used for prediction of closed prices. A group was chosen that has reduced the cost function (MSE), and these groups include the Gaussian Mutation function, the Two-Point Crossover function, and the Uniform Selection function.

A fitness value f that is uniformly picked in the interval [f_{min}, f_{max}], where f_{min} and f_{max} are the lowest and highest fitness values in the current population, is known as a fitness uniform selection system (FUSS).

Table 1 — Change the number of population and generations

Population-size	10	10	20	30	40	50	50	60	70	80	100
Generation	20	40	40	60	70	80	90	100	120	140	160

Table 2 — Genetic algorithms functions

Group	Selection Fcn	Crossover Fcn	Mutation Fcn
Group 1	Roulette	Scattered	Adaptive feasible
Group 2	Stochastic uniform	Constraint dependent	Uniform
Group 3	Uniform	Two point	Gaussian
Group 4	Tournament	Heuristic	Constraint dependent
Group 5	Remainder	Two point	Adaptive feasible

Next, a fitness for each individual i_P that is extremely close to f is chosen, and a copy is added to P. Therefore, it is not required for people to become more fit as a result of this technique. Thus, maintaining variety.[28] In Gaussian mutation each gene K_i is mutated with the mutation rate P_n using Eq. 9[(29)]:

$$K_i^{t+1} = K_i^t + N(0, \sigma) \quad \ldots (9)$$

where, $N(0, \sigma)$ represents the normal distribution (mean equal zero and with σ standard deviation. In the classical GAs the mutation rate keeps constant. However, when the varies of mutation rate based on the algorithmic convergence, the method becomes more efficient.[30] Compute the mutation rate is as follows[30]:

$$p_n^d = Pav + \mu \quad \ldots (10)$$

where, Pav is the average mutation rate and is calculated by using this formula

$$Pav = \frac{Pma+Pmi}{2} \quad \ldots (11)$$

$$\mu = 3\ \sigma(\frac{x}{z}), \ldots \mu \text{ €}[0,3\sigma] \quad \ldots (12)$$

where, Pmi and Pma are the minimum and maximum values of the mutation rate, x is the value of the fitness frequency distribution with M generation, and the linking coefficient is represented by z.

A crossover creates two new children, A and B', from two parents, A and B. One-point, two-point, multi-point, and uniform crossover types are all recognized. In order to achieve prediction, a two-point crossover was applied in this study, which first chooses two random points before exchanging or changing the bit strings between these two positions as shown in Fig. 7.

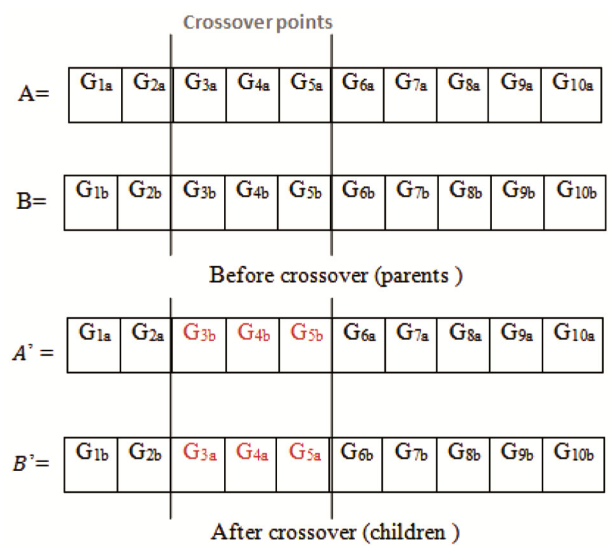

Fig. 7 — Two-point crossover operation

Experimental Results and Discussion

The predictive model's design was analyzed in this work using MATLAB (R2014a), yielding both numerical and graphical findings. The investigation was done with Windows 7 and a Core i5-3230M CPU running at 2.6GHz with 4GB of RAM. Three applied models were used in three steps to generate the results. Finding the general properties of a hybrid model combining neural networks and evolutionary algorithms for four datasets is the first step. Currently, data is being used by Padico to identify the optimal group of functions selection Fcn, Crossover Fcn, and Mutation Fcn that provides the highest performance for GAs, as well as the best parameters for population size, generations, and crossover and migration fractions. The RNNs-LM model is applied for each dataset in the second stage, and the MLPNNs-LM approach is applied in the third stage. A number of neurons in each stage has been changed in each dataset incrementally, starting from 5 neurons to 50 neurons with adding five neurons at a time. In order to

Table 3 — Compares the results of applying the hybrid models (MLPNNs-GAs) with changing functions of GAs

Selection Fcn	Crossover Fan	Mutation Fcn	Mse-Train	Mse-Test
Roulette	Scattered	Adaptive feasible	0.0013	0.0087
Stochastic uniform	Constraint dependent	Uniform	0.0011	0.008
Uniform	Two point	Gaussian	0.0011	0.0051
Tournament	Heuristic	Constraint dependent	0.0013	0.0059
Remainder	Two point	Adaptive feasible	0.0013	0.0052

Table 4 — Compares the results of applying the hybrid models (MLPNNs-GAs) with changing populations and generations

Number of generations	Number of Population	MSE-Train	MSE-Test	Number of generation reached
20	10	0.0011	0.0081128	20
40	10	0.0012	0.0013	40
40	20	0.0008	0.0013	40
60	30	0.0008	0.0012	60
70	40	0.0007	0.0008	70
80	50	0.0008	0.0008	51
90	50	0.0009	0.0013	68
100	60	0.0007	0.0009	60
120	70	0.0008	0.0009	82
140	80	0.0007	0.0009	87
160	100	0.0009	0.0014	89

decrease the time complexity of genetic algorithms, the general characteristics of the MLPNNs-Gas model using the dataset for Padico was first defined. Then this definition was applied to all datasets, beginning with the best group that contained the optimal combination of genetic algorithm functions, as illustrated in Table 3.

There are 20 neurons in the experiment, a population size of 10, a generation number of 10, a migration fraction of 0.1, and a crossover percentage of 0.7. The results of the MLPNNs-GAs model demonstrate that, for all groups that were used, Group 2 and Group 3 each had minimal MSE values for training, while Group 3 had minimum MSE values for testing. Group 3 therefore stands for the ideal combination. Then, the experiment was conducted for migration fraction = 0.1, crossover fraction = 0.7, and with the use of the best group (Group 3) that was produced in the preceding step, as indicated in Table 4, in order to establish the optimal generation and population size. The best or least MSE for both the train and test populations with sizes 40 and 70 generations clarified in Table 4. Following that, the MSE for training and testing has not improved due to an increase in population or generations.

To determine the optimal migration fraction a group of GA parameters was used. These parameters were defined as: population size = 40, generations number = 70, neurons number = 20, crossover fraction = 0.1. Top or lowest MSE in both the test and the train when the migration fraction is 0.01 (MSE test = 0.0049, MSE train = 0.0011), as illustrated in Fig. 8. Changing the migration portion caused us to observe fluctuations in the MSE. As shown in Fig. 9, the fourth stage involves figuring out

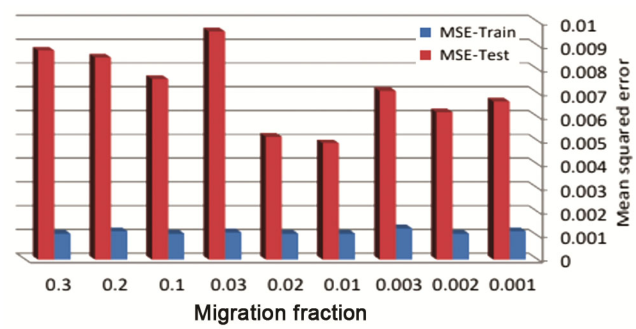

Fig. 8 — Comparison of applying the hybrid models (MLPNNs-GAs) with migration fraction set to (0.001–0.3)

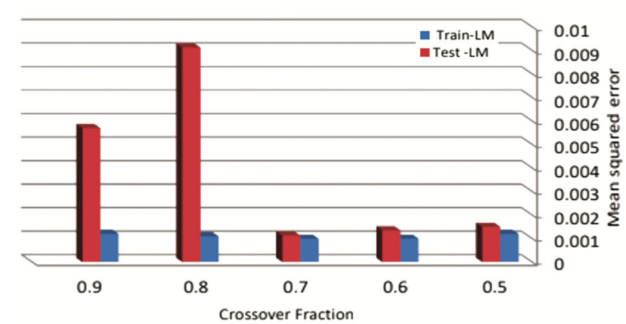

Fig. 9 — Comparison of applying the hybrid models (MLPNNs-GAs) with crossover fraction set to (0.001–0.3)

Table 5 — General characteristics of the hybrid model (MLPNNs-GAs) for 4 datasets

Company Name	Al-Quds Index	Paltel	Padico	Bank of Palestine
Number of generations	70	70	70	70
Number of Population	40	40	40	40
number of neurons	50	50	45	40
Measure of performance	MSE	MSE	MSE	MSE
Selection Fcn	Uniform	Uniform	Uniform	Uniform
Mutation Fcn	Gaussian	Gaussian	Gaussian	Gaussian
Crossover Fcn	Two point	Two point	Two point	Two point
Migration Fraction	0.01	0.01	0.01	0.01
Crossover Fraction	0.7	0.7	0.7	0.7

Table 6 — Results from the hybrid model (MLPNNs-GAs)

Value	Company name	Mse-Train	Mse- Test	Number of generation genetic reached
1912	Padico	0.0005	0.0006	68
384	Al-Quds Index	0.001	0.0011	52
1763	Paltel	0.0005	0.0009	63
1928	Bank of Palestine	0.0008	0.0021	54

Table 7 — Results of MSE for MLPNNs-GAs model for all datasets when we change the number of neurons (5-50)

Company/Index	Padico		Paltel		Bank of Palestine		Al-Quds Index	
No of Neurons	MSE -Train	MSE -Test	MSE -Train	MSE -Test	MSE -Train	MSE -Test	MSE -Train	MSE -Test
5	0.0032	0.0081	0.0032	0.0146	0.0063	0.0062	0.0041	0.0171
10	0.0017	0.0079	0.0017	0.0275	0.0033	0.0046	0.0021	0.0027
15	0.0014	0.0079	0.0001	0.0273	0.0019	0.0042	0.0018	0.0026
20	0.0011	0.0011	0.001	0.0263	0.0013	0.0027	0.0018	0.0025
25	0.0009	0.001	0.0007	0.0246	0.0012	0.0025	0.0015	0.0025
30	0.0009	0.0009	0.0006	0.0023	0.0011	0.0023	0.0014	0.2322
35	0.0008	0.0009	**0.0007**	**0.001**	0.001	0.0022	0.0013	0.0023
40	0.0006	0.0006	0.0006	0.0035	**0.0009**	**0.0021**	0.0013	0.0023
45	**0.0006**	**0.0006**	0.0006	0.0013	0.0009	0.0022	0.0011	0.0022
50	0.0006	0.0007	0.0006	0.0011	0.0009	0.0022	**0.0011**	**0.0021**

the optimal crossover percentage. 20 neurons, 70 generations, a migration fraction of 0.01, Group 3, and 40 population sizes are all used in the example. Using crossover fraction = 0.7 (MSE train = 0.001, MSE test = 0.00113) as the optimum MSE in both the train and test, see Fig. 9. Training and testing MSE are not able to be improved after a 0.7 crossover fraction. Following that, the number of neurons was changed from 5–50 by adding 5 neurons at a time, as indicated in Table 5. The population sizes of 40, 70 generations, migration fraction = 0.01, crossover fraction = 0.7, and Group 3 that have minimum MSE for Padico data as general characteristics of the hybrid model (MLPNNs–GAs) for all datasets were considered. And for each dataset, choose the optimal neuron number with the lowest mean square error (MSE) to utilize for prediction the next year. The MSE results from the hybrid model MLPNNs-GAs are shown in Table 6.

As demonstrated in Table 6, genetic attained is smaller than the determined generation number (70) for each used dataset, indicating that genetic reached the optimal solution and no need to raise generation number.

MLPNNs-GAs

The Padico, Paltel, Bank of Palestine, and Al-Quds Index data sets are the four on which the first model (MLPNNs-GAs) was used. The outcomes of using the MLPNNs-GAs on all of these data sets, varying the number of neurons (5–50) for each dataset are given in Table 7. As can be seen in Table 7, Padico Company's testing error has improved as the number of neurons increases. At 45 neurons, the minimal MSE for both training and testing is reached (MSE train = 0.0006, MSE test = 0.0006). After 45 neurons, the testing error increases even more. Paltel Company found that increasing the number of neurons from 5 to 35 produced a significant improvement in testing

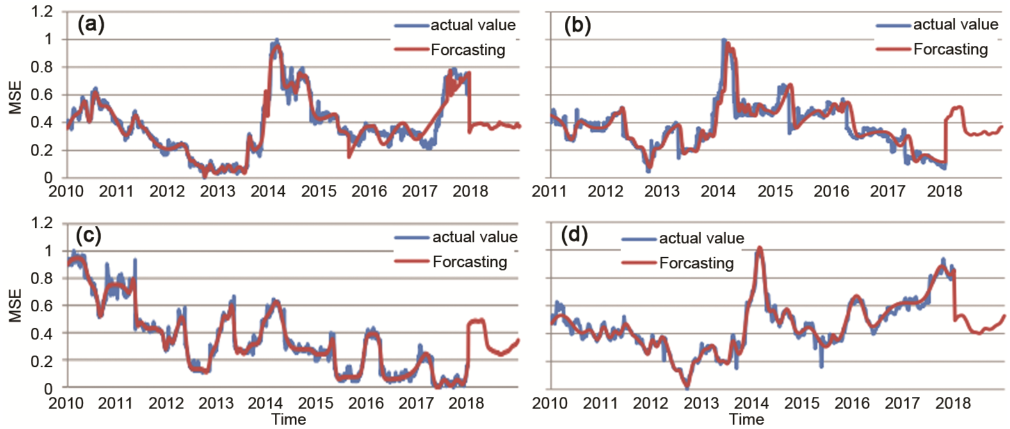

Fig. 10 — Real and predicted values for companies for the next year using (MLPNNs GAs) model: (a) Padico Company, (b) Paltel Company, (c) Bank of Palestine Company, and (d) Al-Quds- Index

Table 8 — Results of MSE for MLPNNs-LM model for all data sets when we change the number of neurons (5-50)

Company/Index	Padico		Paltel		Bank of Palestine		Al-Quds Index	
No of Neurons	MSE - Train	MSE - Train	MSE - Train	MSE -Test	MSE -Train	MSE -Test	MSE - Train	MSE -Test
5	0.0059	0.0061	0.0037	0.0044	0.0129	0.0132	0.0113	0.022
10	0.0019	0.0021	0.0028	0.0033	0.0046	0.0049	0.0042	0.0063
15	0.0017	0.0018	0.0013	0.0016	0.0029	0.0035	0.004	0.0043
20	0.0013	0.0015	0.0011	0.0012	0.0011	0.0024	**0.0021**	**0.0022**
25	0.0013	0.0014	0.0011	0.0012	0.0012	0.0026	0.0015	0.0022
30	0.0009	0.0012	0.0012	0.0012	0.0011	0.0025	0.0014	0.0025
35	0.0007	0.0013	0.0009	0.0011	0.0013	0.0024	0.0016	0.0023
40	0.0008	0.0008	0.0007	0.0013	0.0011	0.0025	0.0017	0.0034
45	0.0007	0.0009	0.0006	0.0009	**0.0012**	**0.0023**	0.0011	0.0023
50	**0.0006**	**0.0007**	**0.0006**	**0.0006**	0.0009	0.0023	0.0011	0.0022

error. Additionally, the MSE test gets worse after the best testing error at 35 neurons (MSE train = 0.0007, MSE test = 0.001). Additionally, add more neurons to the Bank of Palestine Company until 40 neurons significantly boost MSE performance. Testing errors are worse after that. At 40 neurons, the best results were obtained for testing and training (MSE train = 0.0009, MSE test = 0.0021). Increasing the number of neurons improves the performance function for the Al-Quds Index and the optimal performance (lowest training and testing mean square error) at 50 neurons. (MSE test = 0.0021; MSE train = 0.0011). The projected stock prices for the upcoming year of 2018 are shown in Figs. 10.

The stability of the Padico stock price prediction for the upcoming year is displayed in Fig. 10(a). According to the pattern of Padico stock prices over the previous eight years, this suggests that a stable pricing for the upcoming year around the mean stock price of Padico, which equals (mean = 0.378), away from any unstable political situation that might occur in the near future.

Based on Fig. 10(b), it can be inferred that Paltel stock prices will rise in the upcoming year, then fall at the start of the second half of 2018 to reach stable values that will be lower than the average of Paltel stock prices over the last seven years, which is 0.369.

As illustrated in Fig. 10(c), the Bank of Palestine's stock prices will rise at the start of the following year. Converge at stable prices that will surpass the 0.345 average of the bank's stock price over the preceding eight years. Then, stock prices will decline before rising once more at the end of the year.

As can be seen in Fig. 10(d), the Al-Quds-Index stock prices decline at the start of 2018, then level off at the 0.456 mean of the previous eight years' prices, and then rise at the end of the year. MLPNNs-LM is employed on the Padico, Paltel, Bank of Palestine, and Al-Quds Index datasets for these studies as demonstrated in Table 8. It is demonstrated that for

Table 9 — Results of MSE for the RNN-LM model for all data sets when we change the number of neurons (5-50)

Company/Index	Padico		Paltel		Bank of Palestine		Al-Quds Index	
No of Neurons	MSE-Train	MSE -Test	MSE -Train	MSE -Test	MSE -Train	MSE -Test	MSE -Train	MSE-Test
5	0.0040	0.0046	0.0039	0.0041	0.011	0.0129	0.0166	0.0182
10	0.0022	0.0024	0.0046	0.0055	0.0031	0.0032	0.0080	0.0092
15	0.0036	0.0042	0.0034	0.0036	0.0024	0.0021	0.0048	0.0049
20	0.0021	0.0023	0.0023	0.0027	0.0113	0.0131	**0.0037**	**0.0038**
25	0.0035	0.0036	0.0024	0.0029	0.0112	0.0133	0.0056	0.0066
30	**0.0013**	**0.0014**	0.0046	0.0048	0.0019	0.0022	0.0045	0.0045
35	0.0036	0.0037	0.0046	0.0041	0.0063	0.0061	0.0042	0.0043
40	0.0036	0.0041	0.0038	0.0039	0.0030	0.0030	0.0159	0.1593
45	0.0034	0.004	0.0026	0.0034	0.0123	0.0129	0.0047	0.0047
50	0.0036	0.0038	**0.0023**	**0.0024**	**0.0010**	**0.0010**	0.0047	0.0047

Padico, adding neurons increases the testing error by a small amount up to 25 neurons and the training error by a small amount up to 35 neurons. The minimum mean square error (MSE train = 0.00068, MSE test = 0.00079) was found at 50 neurons. For the Paltel Company, increasing the number of neurons results in a good improvement in MSE, with 50 neurons yielding the best error for both training and testing (MSE train = 0.00066, MSE test = 0.00067). In the Bank of Palestine company, increasing the number of neurons causes a variation in the MSE, and the best error for training and testing was obtained with 45 neurons (MSE train = 0.001, MSE test = 0.0023). Furthermore, an increase in neurons for the Al-Quds Index causes variations in the performance function's behavior. peak output with 20 neurons. MSE test = 0.0022, MSE train = 0.0021. According to Table 8, Padico's stock price will be consistent in 2018 by circling around the mean stock price. The results produced are in good agreement with the MLPNNs-GAs results, which are convergent with Padico Company's actual and projected values as well as its forecast for the upcoming year. Furthermore, an increase in neurons for the Al-Quds Index causes variations in the performance function's behavior. peak output with 20 neurons. MSE test = 0.0022, MSE train = 0.0021. According to Table 8, Padico's stock price will be consistent in 2018 by circling around the mean stock price. The results produced are in good agreement with the MLPNNs-GAs results, which are convergent with Padico Company's actual and projected values as well as its forecast for the upcoming year.

Paltel's stock price forecast for 2018 indicates that prices would rise rather than fall by year's end, and this pattern is consistent with the MLPNNs-GAs model's estimate of Bank of Palestine stock prices. The MLPNNs-GAs model predicts that the Al-Quds Index stock prices will decline at the start of 2018 and then increase at the end of the year, following a similar pattern. The Padico, Paltel, Bank of Palestine, and Al-Quds Index data sets were also subjected to the third model (RNN-LM), as demonstrated in Table 9. An increase in neurons does not result in a minor improvement in the MSE for the Padico Corporation. Additionally, 30 neurons produced the best error for testing and training (MSE test = 0.0014, MSE train = 0.0013). Additionally, it shows that when the number of neurons increases for the Paltel Corporation, the enhancing performance function varies and that 50 neurons yield the greatest results for testing and training (MSE train = 0.0023, MSE test = 0.0024). The behavior of MSE performances varies for Bank of Palestine Company as the number of neurons increases. Additionally, 50 neurons produced the best error for testing and training (MSE test = 0.00107, MSE train = 0.00106). The best instruction and evaluation MSE for Al-Quds Index at 20 neurons (MSE train = 0.00379, MSE test = 0.00384) shows that performance fluctuates as the number of neurons increases and continues to fluctuate as 20 neurons become less functional. The values of the actual and anticipated Paltel Company are close to one another but not as accurate as the MLPNNs-GAs and MLPNNs-LM models; RNNs-LM is unable to create good convergence in the actual and predicted values. Furthermore, it is evident that the real and expected values of the Bank of Palestine and the Al-Quds index are somewhat closer to one another.

Result Comparison

As indicated in Tables (7, 8, and 9) in the experimental findings section, a comparative analysis in this part was conducted based on the behavior of MSE errors for all applicable models for each dataset

Table 10 — Min MSE-Errors from each model applied on the same dataset

Company/Index	Padico		Paltel		Bank of Palestine		Al-Quds Index	
Model name	MSE-Train	MSE -Test	MSE –Train	MSE -Test	MSE -Train	MSE -Test	MSE -Train	MSE -Test
MLPNNs-LM	With 50 Neurons		With 50 Neurons		With 45 Neurons		With 20 Neurons	
	0.0006	0.0007	0.0006	0.0006	0.001	0.0023	0.0021	0.0022
MLPNNs-GAs	With 45 Neurons		With 35 Neurons		With 40 Neurons		With 50 Neurons	
	0.0006	0.0006	0.0007	0.001	0.0009	0.0021	0.0011	0.0021
RNN_LM	With 30 Neurons		With 50 Neurons		With 50 Neurons		With 20 Neurons	
	0.0013	0.0014	0.0023	0.0024	0.002	0.0021	0.0037	0.0038

(Padico Company, Paltel Company, Bank of Palestine Company, and Al-Quds index). Using the RNNs-LM model, it is observed that there is variation in the behavior of MSE errors for all companies. The behavior of each dataset is as follows: The behavior of MLPNNs as measured by the Al-Quds index is similar to that of MLPNNs-GAs models; Padico's use of MLPNNs-GAs has outperformed MLPNNs as indicated by the results; and the Bank of Palestine's use of MLPNNs-GAs has outperformed the behavior that results.

How the suggested hybrid model forecasts the closed price better than the MLPNNs-LM and RNNs-LM models is illustrated in Table 10, when the minimum MSE from each model applied to the same dataset is chosen. This is due to the fact that the model chooses GAs rather than RNNs-LM as the optimal initial parameters for NNs, resulting in higher accuracy. The dynamic stock system's patterns could alter since the RNNs-LM model forecasts the future based on historical data.

Conclusions

This research provides a hybrid model MLPNNs-GAs that combines GAs with NNs to predict stock prices on the Palestinian Stock Exchange. The performance of the hybrid model, which optimizes the weights for MLPNNs using GAs, is examined in this paper. The hybrid model was compared with MLPNNs and RNNs models to determine which one would get the best results for forecasting the Palestinian Stock Exchange. According to the study's findings, the hybrid model is a more effective substitute technique for predicting changes in Palestine's stock prices. The predicting performance was significantly enhanced by utilizing hybrid models of MLPNN-GAs, but not with the varied time lags of this index. It's unclear if the hybrid models can function well in volatile markets. In the future, this model will be applied to more stock exchange businesses using various categorization techniques. On the other hand, neuro-fuzzy systems and metaheuristic optimization algorithms can be used to improve the hybrid model performance.

References

1 Lamberton D & Lapeyre B, *Introduction to stochastic calculus applied to finance*, (Chapman and Hall/CRC), (2011), doi: https://doi.org/10.1201/9781420009941.

2 Cao L, AI in finance: challenges, techniques, and opportunities, *ACM Comput Surv*, **55(3)** (2022) 1–38, doi: https://doi.org/10.1145/3502289.

3 Arashi M & Rounaghi M, Analysis of market efficiency and fractal feature of NASDAQ stock exchange: time series modeling and forecasting of stock index using ARMA-GARCH model, *Future Bus J*, **8(1)** (2022) 1–12, doi: https://doi.org/10.1186/s43093-022-00125-9.

4 Amanulla B, Jebran K B & Kohers L S, Efficient capital markets: Are view of theory and empirical work on investors' trading, market timing, and implementation shortfall, *Adv J Fin Innov Rept*, **4(4)** (2020) 1–15.

5 Reddy V K S & Sai K, Stock market prediction using machine learning, *Int Res J Eng Technol*, **5(10)** (2018) 1033–1035.

6 Hamdan I, Awad M & Sabah W, Short-term forecasting of weather conditions in Palestine using artificial neural networks, *J Theor Appl Inf Technol*, **96(9)** (2018).

7 Milana C & Ashta A, Artificial intelligence techniques in finance and financial markets: A survey of the literature, *Strateg Change*, **30(3)** (2021) 189–209, doi: https://doi.org/10.1002/jsc.2403.

8 Kamalov F, Forecasting significant stock price changes using neural networks, *Neural Comput Appl*, **32(23)** (2020) 17655-17667, doi: https://doi.org/10.1007/s00521-020-04942-3.

9 Makridakis S, Spiliotis E & Assimakopoulos V, Statistical and machine learning forecasting methods: Concerns and ways forward, *PloS One*, **13(3)** (2018) e0194889, doi: https://doi.org/10.1371/journal.pone.0194889.

10 Tay F E & Cao L J, Modified support vector machines in financial time series forecasting, *Neurocomputing*, **48(1–4)** (2002) 847–861, doi: https://doi.org/10.1016/s0925-2312(01)00676-2.

11 Lahmiri S, Prediction of international stock markets based on hybrid intelligent systems, In *Handbook of research on innovations in information retrieval, analysis, and management (IGI Global)*, (2016) 110–124, doi: https://doi.org/10.4018/978-1-4666-8833-9.ch004.

12 Boyacioglu M A & Avci D, An adaptive network-based fuzzy inference system (ANFIS) for the prediction of stock market return: The case of the Istanbul stock exchange, *Expert Syst Appl*, **37(12)** (2010) 7908–7912, doi: https://doi.org/10.1016/j.eswa.2010.04.045.

13 Hu H, Tang L, Zhang S & Wang H, Predicting the direction of stock markets using optimized neural networks with Google Trends, *Neurocomputing*, **285** (2018) 188–195, doi: https://doi.org/10.1016/j.neucom.2018.01.038.

14 Aamodt T &Torresen J, Comparing neural networks for predicting stock markets, *Int Conf Eng Appl Neural Net* (Springer), (2017) 363–375, doi: https://doi.org/10.1007/978-3-319-65172-9_31.

15 Chhajer P, Shah M & Kshirsagar A, The applications of artificial neural networks, support vector machines, and long–short term memory for stock market prediction, *Decis Anal J*, **2** (2022) 100015, doi: https://doi.org/10.1016 /j.dajour.2021.100015.

16 Pradeepkumar D & Ravi V, Forecasting financial time series volatility using particle swarm optimization trained quantile regression neural network, *Appl Soft Comput*, **58** (2017) 35–52, doi: https://doi.org/10.1016/j.asoc.2017.04.014.

17 Nayak S C, Misra B B & Behera H S, On developing and performance evaluation of adaptive second order neural network with GA-based training (ASONN-GA) for financial time series prediction, *Proc Advancements Applied Metaheuristic Computing*, IGI global, (2018) 231–263, doi: https://doi.org/10.4018/978-1-5225-4151-6.ch010.

18 Ouyang Z Z, Dedication in online collaboration redeems experience: An analysis on the comparison between Wikipedia and Scholarpedia, *13th Int Symp Distrib Comput Appl Bus, Eng Sci (IEEE),* (2014) 102–106, https://doi.org/ 10.1109/dcabes.2014.24.

19 Wang J Z, Wang J J, Zhang Z G & Guo S P, Forecasting stock indices with back propagation neural network, *Expert Syst Appl*, **38(11)** (2011) 14346–14355, doi: https://doi.org/10.1016/j.eswa.2011.04.222.

20 Atsalakis G S, Protopapadakis E E & Valavanis K P, Stock trend forecasting in turbulent market periods using neuro-fuzzy systems, *Oper Res*, **16** (2016) 245–269, doi: https://doi.org/10.1007/s12351-015-0197-6.

21 Dwaikat M I & Mohammed A, Hybrid model for coronary artery disease classification based on neural networks and evolutionary algorithms, *J Inf Sci Eng*, **38(5)** (2022) 1001–1020.

22 Qteat H & Awad M, Using hybrid model of particle swarm optimization and multi-layer perceptron neural networks for classification of diabete, *Int J Intell Eng Syst*, **14** (2021) 11–22, doi: https://doi.org/10.22266/ijies2021.0630.02.

23 Awad M & Zaid-Alkelani M, Prediction of water demand using artificial neural networks models and statistical model, *Int J Intell Syst Appl*, **11(9)** (2019) 40, doi: https://doi.org/10.5815/ijisa.2019.09.05.

24 Cachier P & Pennec X, 3D non-rigid registration by gradient descent on a Gaussian-windowed similarity measure using convolutions, *IEEE Work Math Meth Bio Imag Anal* (IEEE), (2000) 182–189.

25 Talahmeh A, Awad M & Eleyat M, Time series prediction of server workload using hybrid model of recurrent neural network and genetic algorithms, *Int J Eng Sci*, **10(12)** (2021) 1–10.

26 Liu Y, Novel volatility forecasting using deep learning–long shortterm memory recurrent neural networks, *Expert Syst Appl*, **132** (2019) 99–109, doi: https://doi.org/10.1016/ j.eswa.2019.04.038.

27 Awad M, Optimizing the topology and learning parameters of hierarchical RBF neural networks using genetic algorithms, *Int J Appl Eng Res*, **13(10)** (2018) 8278–8285.

28 Hutter M & Legg S, Fitness uniform optimization, *IEEE Trans Evol Comput*, **10(5)** (2006) 568–589, doi: https://doi.org/10.1109/ tevc.2005.863127.

29 Nelles O & Nelles O, *Nonlinear dynamic system identification*, (Springer), (2020) 831–891, doi: https://doi.org/10.1007/978-3-030-47439-3_19.

30 Blum S, Puisa R, Riedel J & Wintermantel M, Adaptive mutation strategies for evolutionary algorithms, *Annu Conf: EVEN at Weimarer Optimierungsund Stochastiktage*, 2 (2001).