

The Use of Artificial Neural Network in Option Pricing: The Case of S&P 100 Index Options

Zeynep İltüz Samur and Gül Tekin Temur

Abstract—Due to the increasing and varying risks that economic units face with, derivative instruments gain substantial importance, and trading volumes of derivatives have reached very significant level. Parallel with these high trading volumes, researchers have developed many different models. Some are parametric, some are nonparametric. In this study, the aim is to analyse the success of artificial neural network in pricing of options with S&P 100 index options data. Generally, the previous studies cover the data of European type call options. This study includes not only European call option but also American call and put options and European put options. Three data sets are used to perform three different ANN models. One only includes data that are directly observed from the economic environment, i.e. strike price, spot price, interest rate, maturity, type of the contract. The others include an extra input that is not an observable data but a parameter, i.e. volatility. With these detail data, the performance of ANN in put/call dimension, American/European dimension, moneyness dimension is analyzed and whether the contribution of the volatility in neural network analysis make improvement in prediction performance or not is examined. The most striking results revealed by the study is that ANN shows better performance when pricing call options compared to put options; and the use of volatility parameter as an input does not improve the performance.

Keywords—Option Pricing, Neural Network, S&P 100 Index, American/European options

I. INTRODUCTION

PARALLEL with the increasing importance of derivatives in the world financial markets, many pricing techniques have been developed in order to meet the need of estimation of true value. Black and Scholes option pricing is the most renowned formula [1]. Since the formula was developed, researchers have tried to improve the Black-Scholes formula by attacking to the assumptions of the model. For instance, Merton remodelled the Black-Scholes formula by allowing the big price changes with incorporating the jump-diffusion process into the model [2]. Hull and White dealt with the constant variance assumption and modelled the variance as a stochastic process, which allows the changes in time [3]. In addition to modelling variance as a variable that changes in time, Amin and Ng have taken the model one step further by modelling interest rate as a variable as well [4].

Zeynep İltüz Samur is with the Department of Finance, Faculty of Management, Istanbul Technical University, Istanbul, Turkey. (e-mail: iltuzer_zeynep@yahoo.com, iltuzerz@itu.edu.tr).

Gül Tekin Temur is with the Department of Production and Marketing, Faculty of Management, Istanbul Technical University, Istanbul, Turkey. (e-mail: temurg@itu.edu.tr).

Also Duan, Duan and Zhang, Scott and many others have worked on option pricing with changing volatility and/or changing interest rate [5]-[6]-[7].

Artificial Neural Network (ANN) technique has also drawn attention of many researchers in the option pricing topic. Many researches that analyze ANN performance from different perspective have been carried out. In this section we provide an overview of these studies and their findings. Malliaris and Salchenberger examined the performance of neural network option prices with the Black-Scholes prices by using S&P 100 index options [8]. Approximately for half of the cases that they examined, mean squared error for the neural network is smaller than that of Black-Scholes, which implies the good performance of ANN relative to Black and Scholes. Hutchinson, Lo and Poggi studied whether artificial neural network can be used for pricing option in replace of Black and Scholes model with S&P 500 index options [9]. He reported that when parametric methods failed, nonparametric learning –network alternatives can be useful substitutes, but they emphasized that the study did not claim that the learning –network alternatives would be successful in general.

Yao, Li and Tan reported the forecasting performance of back propagation neural network with Nikkei 225 Index futures data [10]. They grouped the data differently to feed the neural network analysis in order to find best combination of input. They also do not take the volatility as an input in the neural network model, but they provide the volatility to be captured by the neural network. They concluded that the grouping data differently creates varying degree of accuracy, and neural network option pricing outperforms the Black-Scholes for high volatile markets.

Amilon studied whether Multi Layer Perceptron (MLP) neural network can be used to find a call option pricing formula better than Black Scholes option pricing formula [11]. Amilon extended the Hutchidson, Lo and Poggio's nonparametric approach and also modelled the spread between bid and ask price by neural network instead of taking the average of bid and ask price simply. Amilon made the performance comparison with two benchmarks, which are Black-Scholes prices with historical volatility and implied volatility. He reported neural networks models outperform either benchmark both in pricing and hedging performances. By working on the Australian Stock Price Index, which also includes American Type option, Daglish reported that neural network analysis showed superior performance for in-sample pricing, however, the parametric

methods showed a better performance in explaining the future prices and showed higher hedging performance [12].

Bennell and Sutcliffe also compared the Black-Scholes performance with artificial neural network (ANN) in pricing European type FTSE 100 call options [13]. They reported that for the out of money options, ANN have unarguable superior performance over Black and Scholes model, but when moving to in-the-money options, performance of Black-Scholes is much better than ANN. However, they also reported that if input data exclude the options with moneyness greater than 1.15 and smaller than 0.9; and maturity greater than 200 days and smaller than 14, then both ANN and Black-Scholes show the same performance.

Anders, Korn and Schmitt examine the artificial neural network model to call options written on the German stock index DAX in order to determine the right combination of input for the best out-of-sample performance by applying the statistical inferences methods [14]. The application of these statistical inferences methods provides a protection against the over-parameterization. They found that the index level improve the out-sample performance of neural network analysis when used in connection with a historical volatility estimate, but when implied volatility is used then the index level shows no improvement in the network performance.

Garcia and Gençay also work on the how pricing accuracy can be improved by homogeneity hint [15]. Instead of setting up a learning network mapping moneyness and maturity directly into the derivative pricing, they break down the pricing function two parts: one with moneyness, the other with the time to maturity. The results of their study showed that the homogeneity hint always reduces the out-of sample performance.

The organization of the paper is as follows: Section 2 gives brief explanation of ANN. Third section details the implementation of ANN. And in the last section, we present the findings of the study.

II. ARTIFICIAL NEURAL NETWORK

ANNs are information processing models that are developed by inspiring from the working principles of human brain. The most essential property of ANN models is its ability of learning from sample sets. There are different kinds of layers in a typical architecture of an ANN model. The basic process units of ANN architecture are neurons which are internally in connection with neurons from subsequent layers. The ability of ANNs to process depends on these connections which are named as weights. The weights give the abilities of prediction or classification to the system. Firstly, the inputs are weighted and summed up. Then they are entered to the activation function in order to get an output from each neuron [16]-[17]. The weights are iteratively changed according to learning rules' results. Consequently, the connections are modified until the best loads are obtained [18].

One of the most commonly used types of ANNs is Multi Layered Perception (MLP) which consists of series of three types of layers with different number of neurons. Input layer is the input receiver from external environment of the

network and output layer includes neurons which transmit the outputs to the decision makers. The third type of layers is named as hidden layer which links inputs to outputs as a black box [19]-[20]. The basic architecture of a MLP network model is showed in Fig. 1 (X_i refers to input values and X_j refers to output values).

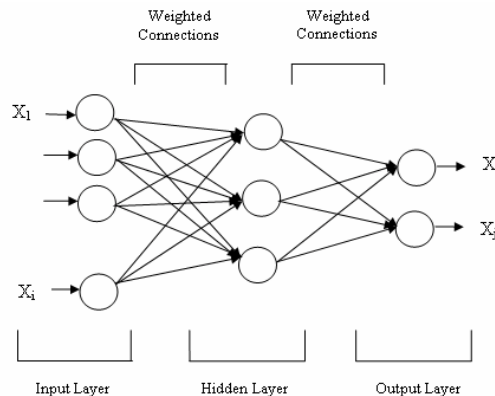


Fig. 1 Architecture of MLP

Many trials are required for deciding on the best loads, best numbers of hidden layers and best numbers of neurons in each layer. However, in literature some rules are also defined to find the best numbers of hidden layers, such as $n/2$, $2n/3$, $n+1$ and $2n+1$ (n is the number of input nodes). Previous researches also show that an ANN model with one hidden layer is sufficient for complex systems. Meanwhile, the number of neurons in each hidden layer should not be chosen larger. Because, larger number of neurons make system memorize the data. Memorization of the system causes to have high error in test results and decrease the generalizability of test results [21].

In practical use, ANNs give many advantages to the decision makers. They do not require any modelling or programming for matching inputs to outputs. ANNs learn relationships of inputs and outputs from sample sets and if the structure is chosen well, the results can be generalized to other data sets. Moreover, they are able to be run with missing or larger data. It is also easy, cheaper and quick to make the system learn from complex data set by training. In consideration of these kinds of advantages, ANNs are used in a wide range of applications in engineering and management practices.

III. ANN IMPLEMENTATION

This part of the study includes the implementation of ANN models in order to predict market prices of call and put options according to variables that will be explained into the following section. Then the results of system will be compared to see how they are close to market data. Basically, it is aimed to give insights into two main research items:

- To develop three different types of MLP network models for the market price prediction process of put and call options.
- To analyze the success of different ANN models in option pricing.

Sub sections of ANN implementation as follows: definition of data set and variable set, network design, network implementation and findings of ANN implementation. First of all, the variables will be identified and then implementation process will be explained.

A. Definition of Data Set and Variable Set

There are 134 data in sample set for both of put and call options. The data cover the quotes covering S&P 100 European and American index options for both put and call type, which are traded on the Chicago Board Options Exchange. The data are obtained from the Wall Street Journal web site. The data includes the options with varied strike prices and maturities on different days, i.e. 15th May, 21st May, 20th June, 21st August, 25th September and 29th October in 2007. The sample data are formed in a way that they incorporate the effects of the variations in the variables into the models. For instance, on 15 May 2005, we have totally 17 options data which include different strike price and maturities.

For ANN implementation, variables, which are listed below, are chosen based on the option pricing literature. In option pricing theory, there are several models developed fundamentally based on strike price, spot price, maturity, riskless interest rate, and volatility. Therefore, we choose these variables as the inputs for ANN. However, how the volatility of an asset should be modelled when option pricing model developed is a deeply analyzed topic; and many researchers have proposed different model for volatility. Some assume the volatility as a constant; some incorporate the volatility as a stochastic variable in the corresponding option pricing model. Therefore, three different types of MLP network which takes different set of variables as inputs are created in order to evaluate the effect of the volatility in the option pricing performance of ANN. There are several ways of estimating volatility, one of which is the historical variance. These variables are defined as the following;

Main Variables

- Type of Options (American/European)
- Strike Price
- Spot Price
- Maturity
- Interest Rate

Volatility Variables

- Variance 1
- Variance 2

When one estimates historical variance, he/she has different choices for data set according to the period that his/her study covers. For instance, while one uses the data going back two years, another one can use data going back one year. In order to capture the change in the variance of the asset for different months, the two different variables, i.e. Variance 1 and Variance 2 are estimated. Two different historical variances, which differs in data range used in the estimation of historical variance, are estimated. Variance 1 estimations are obtained from the period that begins from 2007 and ends on the day of options price gathered. For instance, Variance 1 estimation for options on 15th May is obtained from the period that begins from 2007 and ends on 15th May etc. On the other hand, Variance 2 estimations are obtained only from data that cover the month that option price gathered. For instance, Variance 2 estimation for

options on 15th May is obtained from data in May. The difference between two variables is that they reflect the volatility of the corresponding asset for different time periods. While first one incorporates the past months volatility in estimations, the second one incorporates only the corresponding month volatility of the asset. For variance estimation, we use S&P 100 index data between 3rd January of 2007 and 31st October of 2007 obtained from the <http://finance.yahoo.com/>. U.S. treasury securities at one month constant maturity are used for riskless interest rate and obtained from the Federal Reserve web site.

B. Network Design

The aim of the study is to create three different types of MLP network which takes different set of variables as inputs in, and make comparison to demonstrate which kind of model has the highest prediction performance. For the first model (Model 1), only the main variables are taken into account as inputs. At the second model (Model 2), the main variables and first volatility variable (Variance 1) are defined as inputs. Finally, for the third model (Model 3) the main variables and second volatility variable (Variance 2) are chosen as inputs. Because all models are created to predict market price, the output is the same for all types. As a result, with different inputs, three different ANN structures are built for both of put and call options as shown in Fig. 2, 3 and 4.

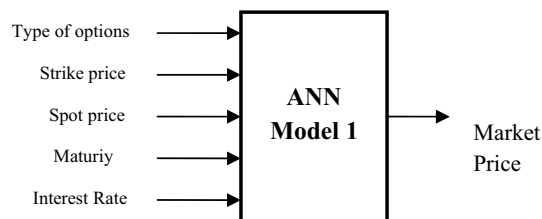


Fig. 2 ANN Structure for Model 1

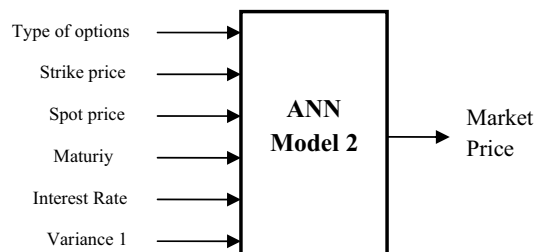


Fig. 3 ANN Structure for Model 2

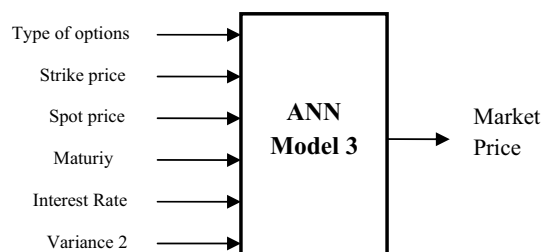


Fig. 4 ANN Structure for Model 3

In the study, three models for call options (Model C1, Model C2, and Model C3) and three models for put options (Model P1, Model P2, and Model P3) are designed. After the definitions of inputs and outputs are accomplished for all models, the data of models are classified for training and testing processes. For both of two options, the sample set with the data of 89 managers is randomly chosen as a training set. It is known that although the error may be seen as decreasing in the training set, it may increase again in the test set. The reason of this situation is that sometimes the network can memorize the training patterns. In order to prevent memorizing, the cross validation set should be formed and cross validation error should be used for evaluation. Therefore, a set for cross validation is formed that consists of 20 options. The rest of the data is kept for testing out sample performance.

C. Network Implementation

Network implementation phase includes training and testing processes. First, the training processes for three models of put and call options are started with a network that has one hidden layer with one neuron. Then, the process is repeated for increasing the number of neurons in the layer up to 15. It is also run for different numbers of hidden layers up to 5 and for different numbers of neurons in each hidden layers up to 15. The selection of an ANN model depends on the value of performance measurement. In this study, Mean Square Error (MSE) is used as a performance measurement. MSE provides effectiveness of the ANN structure. It is formulated as Equation 1 (n is the number of training sample, x_i is the network output and $E(i)$ is the expected value).

$$MSE = \frac{\sum_{i=1}^n (x_i - E(i))^2}{n} \quad (1)$$

It is necessary to choose the number of layers and neurons that make the MSE value minimum. The best models that give the best and minimum MSE values for six types of models are summarized as in Table I and II.

TABLE I
TRAIN AND TEST RESULTS OF BEST NETWORKS
FOR CALL OPTIONS

	TRAIN RESULTS		TEST RESULTS	
	<i>Best Networks (Cross Validation)</i>		<i>Performance Measurement</i>	
Model C1	Hidden Layers (HLs)	2	MSE	2,5124
	Neurons for HL 1 / 2	3 / 3	NMSE	0,0184
	Minimum MSE	0,00088	MAE	1,2193
Model C2	Hidden Layers (HLs)	2	MSE	6,9768
	Neurons for HL 1 / 2	3 / 11	NMSE	0,0147
	Minimum MSE	0,00199	MAE	2,0913
Model C3	Hidden Layers (HLs)	2	MSE	7,8686
	Neurons for HL 1 / 2	2 / 9	NMSE	0,0267
	Minimum MSE	0,00284	MAE	2,0654

TABLE II
TRAIN AND TEST RESULTS OF BEST NETWORKS
FOR PUT OPTIONS

	TRAIN RESULTS		TEST RESULTS	
	<i>Best Network (Cross Validation)</i>		<i>Performance Measurement</i>	
Model P1	Hidden Layers (HLs)	2	MSE	39,1357
	Neurons for HL 1 / 2	3 / 4	NMSE	0,0969
	Minimum MSE	0,00103	MAE	4,1425
Model P2	Hidden Layers (HLs)	2	MSE	18,2177
	Neurons for HL 1 / 2	8 / 4	NMSE	0,3372
	Minimum MSE	0,00288	MAE	3,2440
Model P3	Hidden Layers (HLs)	2	MSE	135,2357
	Neurons for HL 1 / 2	4 / 7	NMSE	0,3402
	Minimum MSE	0,00453	MAE	8,1022

D. Findings of ANN Implementation

As it is seen from Table I and II, all best network structures of call and put options include 2 hidden layers. The smallest neuron number "2" is in the first hidden layer of Model C3, and the largest neuron number "11" is in the second layer of C2. Theoretically, MSE values of training processes which are smaller than 0,01 are found as satisfactory. All MSE values of cross validation sets have acceptable values which are smaller than 0,01. Since these models are examined for testing.

The closeness of test results with market data can be seen from Fig. 5 and 6. The best networks with only main variables are found as closer to the data in market. It reveals that these networks are ready to use for prediction of option market prices. If new values of variables are entered to the models C1 and P1, the decision maker can have an idea about which value the market price will approximately have. On the other hand, especially in Model P2 and Model P3, actual market price and desired market price lines are not overlapped each other. That indicates that these models are inefficient to predict market price, and model results are mostly dissimilar with market data.

Train and test results of best networks for put and call options also give some findings about performance of models. First of all, we found that taking variance as an input does not improve the prediction performance of the ANN, and adding the variance variables in the ANN even worsens the output performance for both put and call options. This can be due to the fact that variance is the only variable whose values are estimated by a method. On the other hand, values of all other variables are observed in the real world and do not require an estimation procedure. Another reason of the relative inefficiency of Model C2, Model C3 and Model P2, Model P3 may depend on the choice of historical variance for volatility estimation. Implied volatility or stochastic models for volatility estimation can improve the performance. However, when two volatility estimates are compared, outperformance of Variance 1 relative to Variance 2 is easily noticed. Therefore, the determination of volatility model for the most efficient ANN can be good future work. Second notable

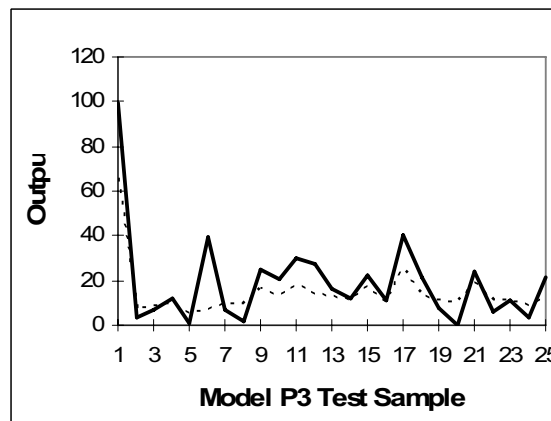
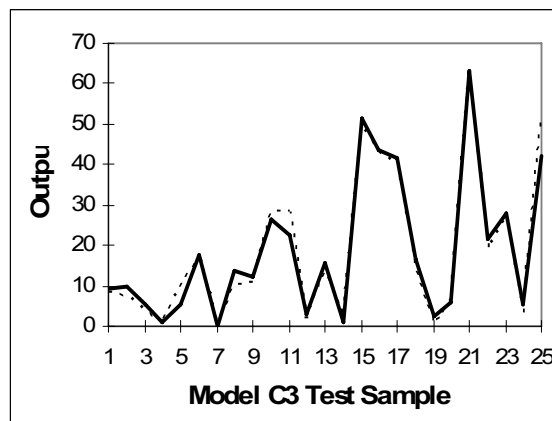
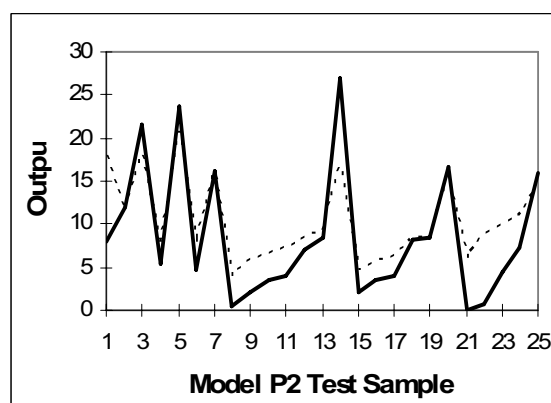
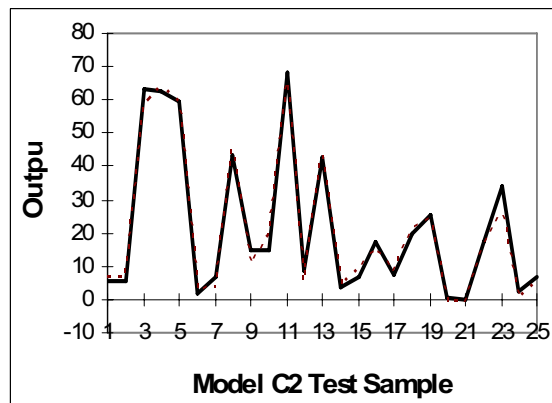
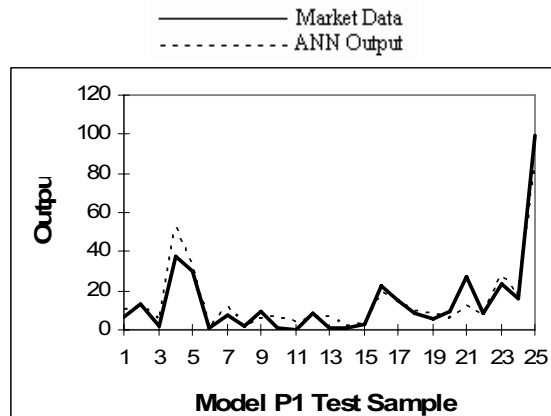
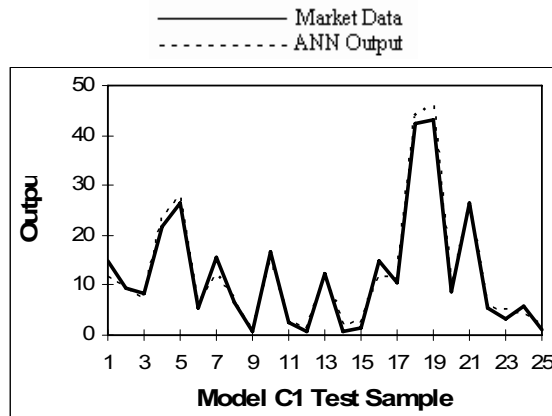


Fig. 5 The closeness of test results with market data for call options

Fig. 6 The closeness of test results with market data for put options

findings of the study are that predictions of call options with ANN have a superior performance than those of put options. This is independent from the data set. Therefore, ANN is not as successful at pricing put options as it is at pricing call options.

In this study, there exists the opportunity to evaluate the performance of ANN in option pricing from different dimensions since the data set includes American/European dimension and moneyness dimension. The Kruskal-Wallis is used to test whether absolute percentage pricing error between market prices and ANN prices for corresponding dimensions are statistically different or not. If the difference is significant

then we can conclude that ANN has varying performance level for American/European dimension or moneyness dimension. Kruskal-Wallis test is chosen since it does not require any assumption such as normality. When the out sample absolute percentage pricing error between market prices and ANN prices of American and European call options is compared by the nonparametric Kruskal-Wallis test, it reveals that ANN shows the same performance for both American and European type. Table III summarizes the examination of ANN performance for American/European dimension and moneyness dimension.

Due to the test result, ANN shows the same performance also for different moneyness level, i.e. out-of-the-money and at-the-money options.

TABLE III
THE KRUSKAL-WALLIS TEST RESULTS FOR
PUT AND CALL OPTIONS*

	American/European	Moneyness
Put	2,04 (0,360)	0,11 (0,744)
Call	3,45 (0,179)	0,46 (0,497)

*Table presents the test results and their probability values in brackets.

IV. CONCLUSION

Purpose of the study is to analyse the success of artificial neural network in put and call option pricing with S&P 100 index data set that is not previously used in terms of the type of the option, i.e. American and European. Briefly the findings are as follows: ANN does not show the same performance at pricing put options as it is at pricing call options with the same input structure. This means that pricing put options with ANN requires extra inputs than those of pricing call options. Also taking historical variance as an input does not improve the performance of ANN for both call and put options. Another finding is that ANN provides the same performance for put and call options no matter the options are American or European type. As a future work, this study can be applied for other options in order to examine the persistence the relatively low pricing performance for put options with a bigger data set, and the study can be improved by taking implied volatility instead of historical volatility in order to measure whether there is an improvement or not in the performance of ANN.

REFERENCES

- [1] F. Black and M. Scholes, "The pricing of options and corporate liabilities," *Journal of Political Economy*, vol. 81, no. 3, pp.637-654, 1973.
- [2] R.C. Merton, "Option pricing when underlying stock returns are discontinuous," *Journal of Financial Economics*, vol. 3, pp. 125-144, 1975.
- [3] J.C. Hull and A. White, "The pricing of options on assets with stochastic volatilities," *The Journal Of Finance*, vol. 42, no. 2, 281-300, 1987.
- [4] K. I. Amin and V. K. Ng, "Option valuation with systematic stochastic volatility," *The Journal of Finance*, vol. 48, no. 3, pp. 881-910, 1993.
- [5] J. Duan, "The Garch option pricing model," *Mathematical Finance*, vol.5, no. 1, pp. 13-32, 1995.
- [6] J. Duan and H. Zhang, "Pricing Hang Seng Index options around the Asian financial crisis- a Garch approach," *Journal of Banking & Finance*, vol. 25, pp. 1989-2014, 2001.
- [7] L. O. Scott, "Pricing stock options in a jump-diffusion model with stochastic volatility and interest Rate," *Mathematical Finance*, vol. 7, no. 4, pp. 413-424, 1997.
- [8] M. Malliaris and L. Salchenberger, "A neural network model for estimating option prices," *Journal of Applied Intelligence*, vol. 3, pp. 193-206, 1993.
- [9] J.M. Hutchinson, A. W. Lo and T., Poggio, "A nonparametric approach to pricing and hedging derivative securities via learning networks," *The Journal of Finance*, vol. 49, no. 3, 851-889, 1994.
- [10] J. Yao, Y. Li and C. L. Tan, "Option price forecasting using neural networks," *Omega*, vol. 28, pp. 455-466, 1999.
- [11] H. Amilon, "A neural network versus Black-Scholes: a comparison of pricing and hedging Performances", *Journal of Forecasting*, vol. 22, pp. 317-335, 2003.
- [12] T. Daglish, "A pricing and hedging comparison of parametric and nonparametric approaches for American Index Options," *Journal of Financial Econometrics*, vol. 1, no. 3, pp. 327-364, 2003.
- [13] J. Bennell and C. Sutcliffe, "Black-Sholes versus artificial neural networks in pricing FTSE 100 options," *Intelligent Systems In Accounting, Finance and Management*, vol. 12, pp. 243-260, 2004.
- [14] U. Anders, O. Korn and C. Schmitt, "Improving the pricing of options: a neural network approach," *Journal of Forecasting*, vol. 17, pp. 369-388, 1998.
- [15] R. Garcia and R. Gençay, "Pricing and hedging derivative securities with neural networks and homogeneity hint," *Journal of Econometrics*, vol. 94, pp. 93-115, 2000.
- [16] S. Kahraman, O. Gunaydin, M. Alber and M. Fener, "Evaluating the strength and deformability properties of misis fault breccia using artificial neural networks," *Expert Systems with Applications*, vol. 36, pp. 6874-6878, 2009.
- [17] P. Gupta and N. K. Sinha, "An improved approach for nonlinear system identification using neural networks," *Journal of the Franklin Institute*, vol. 336, no. 4, pp. 721-734, 1999.
- [18] J. Johnson and P. Picton, "Mechatronics: Designing Intelligent Machines, Concepts in Artificial Intelligence," Butterworth-Heinemann vol. 2, 1995.
- [19] E. Koskivaara, "Neural networks in analytical review procedures," *Managerial Auditing Journal*, vol. 19, no.2, pp. 191-223, 2004.
- [20] K.L. Choya, W.B. Leea and V. Lob, "Design of an intelligent supplier relationship managementsystem: a hybrid case based neural network approach" *Expert Systems with Applications*, vol. 24, pp. 225-237, 2003.
- [21] T.C. Tang and L.C. Chi, "Neural networks analysis in business failure prediction of Chinese importers: A between-countries approach" *Expert Systems with Applications*, vol. 29, pp. 244-255, 2005.

Zeynep ILTUZER SAMUR is a teaching and research assistant at Istanbul Technical University (ITU). She had a BS in mathematical engineering at Yıldız Technical University and an MS in management engineering at ITU. She is preparing her PhD at the same faculty and works on derivatives and risk management.

Gül Tekin TEMUR is a teaching and research assistant at Istanbul Technical University (ITU). She had a BS in management engineering at Istanbul Technical University at Faculty of Management. She is preparing her PhD thesis in production management.