

# A Hybrid Artificial Neural Network Model for Option Pricing

Hanningtone Meshack Simiyu, Anthony Gichuhi Waititu and Jane Aduda Akinyi

*Pan African University Institute for Basic Sciences, Technology and Innovation (PAUISTI), Nairobi, Kenya;  
Department of Statistics and Actuarial Science, Jomo Kenyatta University of Agriculture and Technology (JKUAT),  
Nairobi, Kenya*

## Article history

Received: 16-05-2019

Revised: 03-07-2019

Accepted: 06-08-2019

## Corresponding Author:

Hanningtone Meshack Simiyu  
Pan African University Institute  
for Basic Sciences, Technology  
and Innovation (PAUISTI),  
Nairobi, Kenya; Department of  
Statistics and Actuarial Science,  
Jomo Kenyatta University of  
Agriculture and Technology  
(JKUAT), Nairobi, Kenya  
Email: shanningtone13@gmail.com

**Abstract:** In the absence of a well-defined input selection technique associated with the pure ANN models, Option pricing using pure ANN models while relaxing the assumption of constant volatility remains a challenge. The conservative drill espoused has been to make allowances for a large number of input lags with the confidence that the ability of ANN to integrate suppleness and redundancy generates a more robust model. This is to say that the nonexistence of input selection criteria notwithstanding, the models have been developed without due consideration to the effect that the choice of input selection technique would have on model complexity, learning difficulty and performance measures. In this study, we deviate from the conventional techniques applied by the pure ANN option price models and adopt the hybrid model in which the volatility component is handled using some celebrated time series models, with specificity to the ANN-GJR-GARCH model - a hybrid of the ANN and a time series hybrid. The hybrid ANN option pricing model is then framed and tested with the forecasts of the ANN-GJR-GARCH model as a volatility input alongside two other inputs - time to maturity and moneyness. Finally, we compare the performance of the hybrid model developed with that of a pure ANN model. Results indicate that the hybrid model outperforms the pure ANN model not only in forecasting but also in the training time and model complexity.

**Keywords:** A Hybrid Model, Artificial Neural Networks, Option Pricing

## Introduction

Since the establishment of the Chicago Board of Option exchange in 1973 (Hull *et al.*, 2013), focus was shifted to option valuation methods with the solemn aim of modelling option prices. It was around this time that the Black-Scholes Model (BSM) was developed – a ground-breaking model that symbolized a major breakthrough in the financial sector. Since then and for the past two decades, the model has continuously received considerable attention especially in underlying probability attributes of an European call option on a non-dividend stock (Al Saedi and Tularam, 2018) and has been identified as the basic building block of the financial derivatives theory (Wilmott *et al.*, 1995). Nonetheless, numerous studies have exposed a series of inconsistencies that exist between the market prices and the BSM output - a problem that has directly been linked to the widely disputed assumptions governing the development and use of the BSM, including but not limited to: Constant

volatility, existence of efficient markets, non-dividend paying structures, known and constant interest rates, log-normally distributed returns, commission free transactions and a perfectly liquid market.

In an effort to manage these inconsistencies, advances have been made in both the model itself and its respective solution methods – both analytical (Harper, 1994; Forsyth *et al.*, 1999; Jódar *et al.*, 2005; Rodrigo and Mamon, 2006; Bohner and Zheng, 2009; Edeki *et al.*, 2015; Shin and Kim, 2016) and numerical (Cortés *et al.*, 2005; Company *et al.*, 2008; Ankudinova and Ehrhardt, 2008; Černá *et al.*, 2016). However, in spite of such concerted efforts having been employed to improve the performance of the BSM, multiple studies still show that these techniques have not been able to very well capture more realistic aspects that characterize day-to-day transactions such as the presence of transaction costs, high volatility, illiquid markets and large investor preference. In addition, market participants change their option pricing attitudes from time to time and thus, many

researchers hold the view that existing complexity issues require non-linear solutions.

It is against this background that a good number of researchers have shifted focus to non-parametric models flexible enough and capable of capturing the non-linear patterns that characterize the observed market data. One such model is the Artificial Neural Network (ANN) model. ANNs have increasingly become promising alternatives to the BSM and other parametric option pricing models due to the fact that unlike the BSM, they do not require or rely on any underlying assumptions (Malliaris and Salchenberg, 1993) and are trained inductively using historical or implied input variables and option transaction data (Andreou *et al.*, 2006).

The very first attempt to estimate option prices using ANNs was made by Hutchison *et al.* (1994). They used three different network architectures namely: Radial Basis Function (RBF), Multilayer Perception (MLP) and Projection Pursuit Regression (PPR) to fit both Monte-Carlo simulated Brownian underlier and Black-Scholes option data and S&P 500 futures thereof. The authors however used a minimalist approach in the selection of their inputs and restricted the network inputs to time to maturity (T-t) and Moneyness. Interest rate and volatility were also assumed to be constant. It is also worth noting that the study used financial knowledge in construction, namely the “homogeneity property” of the option price formula which was borrowed from Merton (1990), consequently justifying the use of moneyness instead of the underlying price and strike price separately.

Mostafa and Dillon (2008) compared the option pricing performance of the ANN model to the Black-Scholes and the Generalized Autoregressive Conditional Heteroscedastic (GARCH) pricing models. The study used a MLP with a single layer of hidden nodes. The ANN was trained on the implied volatility rather than the option price and this led to an improved performance compared to the competing models. The hedging performance of the neural network, the GARCH option-pricing model and the Black-Scholes were also analyzed. According to Hajizadeha and Seifia (2011), one of the limitations of the BSM that prompted the application of the ANN is the controversial assumption that the underlying probability distribution is lognormal. Their study thus proposed a couple of hybrid models to reduce these limitations and enhance the ability of option pricing. The key input to their option pricing model was volatility, in which three popular GARCH type models were used in estimating volatility. Two nonparametric models based on neural networks and neuro-fuzzy networks were then developed to price call options for S&P 500 index. Results were then compared with those of Black-Scholes model and they showed that both the neural network and the neuro-fuzzy network models outperformed the Black-Scholes model. Furthermore,

comparing the neural network and neuro-fuzzy approaches, Hajizadeha and Seifia (2011) observed that for At-The-Money (ATM) options, the neural network model performed better and for both In-The-Money (ITM) and Out-of-The Money (OTM) options, the neuro-fuzzy model provided better results. Mitra (2012) using Nifty call option prices, made an attempt to improve accuracy of option price estimation using ANNs by adjusting all input parameters using a suitable multiplier. The values of these multipliers were determined using known data that minimizes errors in valuation.

Another application involving ANNs on option pricing was one done by Andreou *et al.* (2005). The study compared the option pricing ability of Robust ANNs optimized with the Huber function against those optimized with Least Squares. The comparison was in respect to pricing European call options on the S&P 500 using daily data for the period April 1998 to August 2001. In the study, the analysis was augmented with the use of several historical and implied volatility measures. The study also went a step further to include hybrid networks that directly incorporated information from the parametric model in the analysis. It was found that the ANNs modelled with the use of the Huber function outperformed the ones optimized with least squares. Enke and Dagli (2017) applied a hybrid neural network which preprocessed financial input data for improving the estimation of option market prices. The model in this study comprised of two parts. In the first part, a neural network model was developed to estimate volatility, while in the second part an additional neural network was developed to value the difference between the BSM results and the actual market option prices. The resulting option price was then a summation between the BSM and the network’s response. The study obtained that the hybrid system with a neural network for estimating volatility provided better performance in terms of pricing accuracy than either the BSM with historical volatility, or the BSM with volatility valued by the neural network.

Malliaris and Salchenberg (1993), developed an ANN model that processes financial input data to estimate market option prices at closing. The ANNs ability was compared to the BSM, a comparison that revealed that the Mean Squared Error (MSE) for the ANN was less than that of the BSM in more than half the cases examined. The ANN model used exactly the same financial data as the BSM. Amilon (2003) examined whether an MLP ANN, could be used to find a call option pricing formula better corresponding to market prices and the properties of the underlying asset than the Black-Scholes formula. The neural network method was applied to the out-of-sample pricing and delta-hedging of daily Swedish stock index call options from 1997-1999, with the BSM with historical and implied volatility as a benchmark. The findings revealed that the ANN

outperformed the benchmarks in both pricing and hedging (Amilon, 2003). Gradojevic *et al.* (2007) applied a non-parametric modular neural network (MNN) model to price the S&P-500 European call options. The modules were based on time to maturity and moneyness of the options. The option price function of interest was homogenous of degree one with respect to the underlying index price and the strike price. The study found that modularity improved the generalization properties of standard feedforward ANN option pricing models (with or without the homogeneity), relative to the Black-Scholes model.

It is however worth noting that a good number of these studies have curatively dealt with the linearity aspect that existed in the parametric models while still clinging to the same assumptions on the BSM. Hutchnison *et al.* (1994) for instance, by using a minimalist approach in the selection of their inputs and restricting the network inputs to time maturity (T-t) and Moneyness, ends up holding interest rate and volatility constant – the same assumption that draws criticism in the BSM. Secondly, due to lack of a well-defined input variable selection techniques associated with the ANN models, pricing of options using ANN while relaxing some assumptions such as the assumption on constant volatility still remains a challenge. The conventional practice adopted in modelling option prices using pure ANNs has been the trial and error method in which allowances are made for a large number of lags (in the case of volatility) which are then gradually reduced, with the belief that the ability of ANN to incorporate flexibility and redundancy creates a more robust model. This is to say that lack of input variable selection criteria notwithstanding, many models have been developed without due consideration to the effect that the choice of input variable selection techniques would have on model complexity, learning difficulty and performance measures. Furthermore, majority of the studies using pure ANN models have handled the time variant volatility component by pre-modelling it using ANN, before using it as an input to the ANN option pricing model. However and while recognizing the fact that financial time series data harbors a set of characteristics that tend to drive estimation procedures away from linear models, such as volatility clustering, leptokurtosis and leverage effects, it should not pass unnoticed that ANNs in their pure forms have proved inadequate in capturing these characteristics. Consequently, this study sought to deviate from the conventional pure ANN modelling by developing a hybrid model in which an input to the network is externally determined. Here, the volatility component is handled using some celebrated time series models, with specificity to the ANN-GJR-GARCH model - a hybrid of the ANN, the Glosten, Jagannathan and Runkle (GJR) model and the time series GARCH

model. The hybrid ANN option pricing model is then built and tested with the forecasts of the ANN-GJR-GARCH model as a volatility input alongside two other inputs - time to maturity and moneyness. Finally, we compare the performance of the hybrid model developed with that of a pure ANN model.

The contribution of this paper is thus threefold. First, the study provides a remedy on ANN's inadequacy in capturing some features that characterize financial time series data such as volatility clustering, leptokurtosis and the existence of leverage effects. Secondly, the study provides a navigation on the problem of a lack of a well-defined Input Variable Selection (IVS) technique that characterizes ANNs. Finally, the study provides an alternative non-parametric option pricing model consistent with the observed market data and capable of capturing changes in the option pricing attitudes of market participants from time to time.

## Materials and Methods

### Data

The study used intraday data for the AAPL stock option for the period between December 2016 and March 2017 with 56,238 data points. The distribution of these data points was as follows: Training (50%), testing (25%) and validation (25%). This is further explained in detail in the respective model subsections.

### Assumptions and Restrictions

In developing the hybrid ANN model, the assumption on constant volatility was relaxed. However, the following assumptions were upheld: First, that there were no transaction costs, secondly, that there were no dividends during the life of the derivative and finally, that the risk-free rate of interest,  $r$ , was constant and the same for all maturities.

There are also basic conditions that any option pricing model is required to fulfil, among them the requirement that the price outputs be greater than or equal to zero as well as the homogeneity property. These, as clearly articulated in the later sections of the methodology, not only informed the choice of the activation function but also the definition of the inputs, particularly the moneyness input variable.

### Volatility Input

Engle (1982) proposed to model time varying conditional variance by applying the Autoregressive Conditional Heteroscedasticity Process (ARCH) with the expectation of capturing the dynamic behaviors of conditional variance by application of lagged disturbance. Similarly, Engle and Bollerslev (1986) proposed one step ahead forecasting to overcome the vice related to the ARCH models with regards to the number of parameters.

By so doing, the infinite number of parameters in the ARCH model can be reduced to only two parameters in the GARCH model (Najjar *et al.*, 1997).

Both the ARCH and the GARCH models simultaneously capture volatility clustering and leptokurtosis hence their wider use in the financial market analysis. However, they both fail to capture the leverage effects. Nelson (1991) proposed an extended ARCH model, the Exponential Generalized Auto Regressive Conditional Heteroscedasticity (EGARCH) model capable of capturing the asymmetric shock associated with the conditional variance as well as the leverage effects. Glosten *et al.* (1993) modified the models by proposing a GARCH model that performed better than the preceding. Brownlees *et al.* (2011) introduced the GJR model and found it to be the best forecaster among asymmetric model and GARCH for one step and multi-step ahead forecasting. Later on, Lu *et al.* (2016) combined the GJR model and the GARCH model to obtain a GJR-GARCH model. Different studies have found the GJR-GARCH model to outperform the individual models, each separately applied. Thus, in this section, we deviate from the conventional techniques applied in option price modelling using pure ANNs and adopt the hybrid model in which the volatility component is handled using some celebrated time series models, with specificity to the ANN-GJR-GARCH models, also a hybrid of ANN and a time series hybrid of Glosten *et al.* (1993) and that of Engle and Bollerslev (1986). The hybrid ANN option pricing model is then framed and tested with the forecasts of the ANN-GJR-GARCH model as a volatility input alongside time to maturity and moneyness.

Now let  $r_t$  denote the log returns of the price of our underlying asset (the AAPL stock) at the time index  $t$ . To put the GARCH model in a proper perspective, it is substantive to consider the conditional mean and conditional variance of  $r_t$ , given  $I_{t-1}$ , where  $I_{t-1}$  denotes the information available at time  $t-1$ . Typically,  $I_{t-1}$  consists of all linear functions of the past returns. Then we have:

$$\begin{aligned} \mu_t &= E(r_t/I_{t-1}) \\ \sigma_t^2 &= Var(r_t/I_{t-1}) = E\{r_t - E(r_t/I_{t-1})\}^2 \end{aligned} \quad (1)$$

Now let  $\epsilon_t = r_t - E(r_t/I_{t-1})$  be the mean-corrected log return. Then  $\epsilon_t$  follows a GARCH ( $m, n$ ) model if:

$$\begin{aligned} \epsilon_t &= \sigma_t z_t \\ \sigma_t^2 &= \phi_0 + \sum_{i=1}^m \phi_i \epsilon_{t-i}^2 + \sum_{j=1}^n \theta_j \sigma_{t-j}^2 \end{aligned} \quad (2)$$

where,  $\{z_t\}$  is a sequence of independent and identically distributed random variables with mean 0 and variance 1,  $\phi_0 > 0, \phi_i \geq 0, \theta_j \geq 0$  and  $\sum_{i=1}^{\max(m, n)} (\phi_i + \theta_j) < 1$ .

It is also understood herein that  $\phi_i = 0$  for  $i > m$  and  $\theta_j = 0$  for  $j > n$ , with the latter constraint on  $\theta_j, \phi_j$  implying that the conditional variance of  $\epsilon_t$  is finite while the variance  $\sigma_t^2$  evolves over time. If  $n = 0$ , then the Equation 2 reduces to a pure ARCH ( $m$ ) model.

On the other hand, the GJR model is a transformation for the Asymmetric Power ARCH model. The latter is defined as follows: for the mean corrected returns  $\epsilon_t = r_t - E(r_t/I_{t-1})$  such that  $\epsilon_t = \sigma_t Z_t$  where  $\epsilon_t \sim N(0, 1)$ , we have:

$$\sigma_t^\delta = \omega + \sum_{i=1}^q \alpha_i (|\epsilon_{t-i}| - \psi_i \epsilon_{t-i})^\delta + \sum_{j=1}^p \theta_j \sigma_{t-j}^\delta \quad (3)$$

Where:

$$\begin{aligned} \omega &> 0 \\ \delta &> 0 \\ \alpha_i &\geq 0, i = 1, \dots, q \\ -1 < \psi_i < 1, i = 1, \dots, q \\ \theta_j &\geq 0, j = 1, \dots, p \end{aligned}$$

For the GJR model  $\delta = 2$  and we have: When  $\delta = 2$  and  $0 \leq i < 1$ , then:

$$\begin{aligned} \sigma_t^2 &= \omega + \sum_{i=1}^m \alpha_i (|\epsilon_{t-i}| - \psi_i \epsilon_{t-i})^2 + \sum_{j=1}^n \theta_j \sigma_{t-j}^2 \\ &= \omega + \sum_{i=1}^q \alpha_i (|\epsilon_{t-i}|^2 - \psi_i^2 \epsilon_{t-i}^2 - 2\psi_i |\epsilon_{t-i}| \epsilon_{t-i}) + \sum_{j=1}^n \theta_j \sigma_{t-j}^2 \quad (4) \\ &= \begin{cases} \omega + \sum_{i=1}^m \alpha_i (1 + \psi_i)^2 \epsilon_{t-i}^2 + \sum_{j=1}^n \theta_j \sigma_{t-j}^2, \epsilon_{t-i} < 0 \\ \omega + \sum_{i=1}^m \alpha_i (1 - \psi_i)^2 \epsilon_{t-i}^2 + \sum_{j=1}^n \theta_j \sigma_{t-j}^2, \epsilon_{t-i} > 0 \end{cases} \end{aligned}$$

$$\begin{aligned} \sigma_t^2 &= \omega + \sum_{i=1}^m \alpha_i (1 - \psi_i)^2 \epsilon_{t-i}^2 \\ &+ \sum_{i=1}^m \alpha_i \{(1 + \psi_i)^2 - (1 - \psi_i)^2\} S_i \epsilon_{t-i}^2 + \sum_{j=1}^n \theta_j \sigma_{t-j}^2 \end{aligned} \quad (5)$$

$$\begin{aligned} \sigma_t^2 &= \omega + \sum_{i=1}^m \alpha_i (1 - \psi_i)^2 \epsilon_{t-i}^2 + \sum_{j=1}^n \theta_j \sigma_{t-j}^2 + \sum_{i=1}^m 4\alpha_i \psi_i D_i \epsilon_{t-i}^2 \\ D_i &= \begin{cases} 1, \text{if } \epsilon_{t-i} < 0 \\ 0, \text{otherwise} \end{cases} \end{aligned} \quad (6)$$

Defining  $\alpha_i(1 - \psi_i)^2$  and  $4\alpha_i \psi_i = \psi_i^*$ , then we have:

$$\sigma_t^2 = \omega + \sum_{i=1}^m \phi_i \epsilon_{t-i}^2 + \sum_{j=1}^n \theta_j \sigma_{t-j}^2 + \sum_{i=1}^m \psi_i^* D_{t-i} \epsilon_{t-i}^2 \quad (7)$$

From here, the GJR-GARCH model is defined by:

$$\sigma_t^2(G-GARCH) = \omega + \sum_{i=1}^m (\varphi_i + \psi_i^* D_{t-i}) \epsilon_{t-i}^2 + \sum_{j=1}^n \theta_j \sigma_{t-j}^2 \quad (8)$$

$$D_i = \begin{cases} 1, & \text{if } \epsilon_{t-i} < 0 \\ 0, & \text{if } \epsilon_{t-i} \geq 0 \end{cases}$$

In developing the ANN-GJR-GARCH volatility model, the latter is built by considering the volatility output layer of the ANN as a variable of the GJR-GARCH model, defined in 8 above. The output layer of the ANN consists of the AAPL stock returns modelled with the lags of the underlying asset price as the inputs to the ANN model. i.e., if we let  $S_t$  be the AAPL stock price at time  $t$ , then,  $r_t = f(S_{t-1}, S_{t-2}, \dots, S_{t-q})$ . Letting  $\aleph(r_t)$  denote the output layer of ANN stock returns model, then, the ANN-GJR-GARCH model is of the form:

$$\sigma_{t(AGG)}^2 = \omega + \sum_{i=1}^m (\varphi_i + \psi_i^* D_{t-i}) \epsilon_{t-i}^2 + \sum_{j=1}^n \theta_j \sigma_{t-j}^2 + \sum_{k=1}^S \beta_k \aleph(r_t) \quad (9)$$

### The Hybrid ANN Model for Option Pricing

In building the ANN option pricing model, the neural network option pricing model is trained and tested based on the volatility forecasts of the ANN-GJR-GARCH model. Other inputs included are moneyness and time to maturity, which we begin by scaling to values between 0 and 1 to have:

$$\left(\frac{X}{S_t}\right)_\alpha^* = \frac{\left(\frac{X}{S_t}\right)_\alpha - \text{Min}\left(\frac{X}{S_t}\right)_\alpha}{\text{Max}\left(\frac{X}{S_t}\right)_\alpha - \text{Min}\left(\frac{X}{S_t}\right)_\alpha}, \alpha = 1, 2, \dots, n \quad (10)$$

$$(T-t)_\alpha^* = \frac{(T-t)_\alpha - \text{Min}(T-t)_\alpha}{\text{Max}(T-t)_\alpha - \text{Min}(T-t)_\alpha}, \alpha = 1, 2, \dots, n \quad (11)$$

and:

$$\sigma_{t(AGG)\alpha}^{2*} = \frac{\{\sigma_{t(AGG)\alpha}^2\} - \text{Min}(\sigma_{t(AGG)\alpha}^2)}{\text{Max}(\sigma_{t(AGG)\alpha}^2) - \text{Min}(\sigma_{t(AGG)\alpha}^2)}, \alpha = 1, 2, \dots, n \quad (12)$$

The resultant model thus consists of three input neurons, each of which is multiplied by weights and aggregated as inputs to the  $j^{\text{th}}$  hidden node. The inputs are obtained as follows: For Moneyness, we have the transformation:

$$I_{\alpha jM} = \omega_{jM} \left(\frac{X}{S_t}\right)_\alpha^*, j = 1, 2, \dots, H \quad (13)$$

Where:

$I_{\alpha jM}$  = The scaled moneyness input to the  $j^{\text{th}}$  hidden node

$\omega_{jM}$  = Weights connecting the scaled moneyness to the  $j^{\text{th}}$  hidden node  
 $M$  = Index notation for moneyness  
 $j$  = Hidden neuron position

For time to maturity, we have:

$$I_{\alpha j\tau} = \omega_{j\tau} \tau_\alpha^*, j = 1, 2, \dots, H \quad (14)$$

Where:

$I_{\alpha j\tau}$  = Time to maturity input to the  $j^{\text{th}}$  hidden neuron  
 $\omega_{j\tau}$  = Weights connecting the scaled time to maturity neuron to the  $j^{\text{th}}$  hidden neuron

and as for the third input, we have:

$$I_{\alpha j\sigma(AGG)} = \omega_{j\sigma(AGG)} (\sigma_{t(AGG)\alpha}^{2*}) \quad (15)$$

Where:

$I_{\alpha j\sigma(AGG)}$  = The normalized ANN-GJR-GARCH volatility forecast inputs to the  $j^{\text{th}}$  hidden neuron  
 $\omega_{j\sigma(AGG)}$  = Weights connecting the normalized ANN-GJR-GARCH volatility forecasts to the  $j^{\text{th}}$  hidden neuron

An aggregation of these yields the net input to the  $j^{\text{th}}$  hidden node which we obtain as:

$$I_{Hj} = \omega_{j0} + \omega_{jM} \left(\frac{X}{S_t}\right)_\alpha^* + \omega_{j\tau} \tau_\alpha^* + \omega_{j\sigma(AGG)} (\sigma_{t(AGG)\alpha}^{2*}) \quad (16)$$

The sigmoid function is used in rescaling the inputs to the  $j^{\text{th}}$  neuron back to (0,1). This is necessary first, to satisfy the option pricing conditions such as the positivity of option prices and secondly in rescaling the inputs from other neurons back to (0,1). This lead to a transformation of the input  $I_{Hj}$  to the  $j^{\text{th}}$  hidden neuron as follows:

$$Q_j(\omega_{j0}, \omega_{jM}, \omega_{j\tau}, \omega_{j\sigma(AGG)}, X, S_t, \tau, \sigma_{t(AGG)\alpha}^{2*}) = \frac{1}{1 + \exp(-1) \left\{ \omega_{j0} + \omega_{jM} \left(\frac{X}{S_t}\right)_\alpha^* + \omega_{j\tau} \tau_\alpha^* + \omega_{j\sigma(AGG)} (\sigma_{t(AGG)\alpha}^{2*}) \right\}} \quad (17)$$

where,  $Q_j(\omega_{j0}, \omega_{jM}, \omega_{j\tau}, \omega_{j\sigma(AGG)}, X, S_t, \tau, \sigma_{t(AGG)\alpha}^{2*})$  = the output of the  $j^{\text{th}}$  hidden neuron, for  $j = 1, 2, \dots, H$ .

### Determining $H$ , the Number of Hidden Nodes

In determining  $H$ , an approach in which all the three suggestions by Salchenberger *et al.* (1992), Hajela and Berke (1991) and Hecht-Nielsen (1989) and Caudill

(1988) on how to obtain  $H$  were incorporated. This lead to the formula:

$$\text{Min}\left(\frac{I+O}{2}, 0.75I\right) \leq H \leq 2I+1 \quad (18)$$

and since  $I=3$  and  $O = 1$ , we have:

$$\begin{aligned} \text{Min}\left(\frac{3+1}{2}, 0.75(3)\right) &\leq H \leq 2(3)+1 \\ \Leftrightarrow \text{Min}(2, 2.25) &\leq H \leq 7 \\ \Leftrightarrow 2 &\leq H \leq 7 \end{aligned} \quad (19)$$

Where:

- $H$  = Denote the number of hidden neuron
- $I$  = The number of input nodes
- $O$  = The number of output nodes

The net input to the output neuron is then obtained as:

$$\begin{aligned} I_{j0} &= \omega_0 + \sum_{j=1}^H \omega_{0j} Q_j(\omega_{j0}, \omega_{jM}, \omega_{j\tau}, \omega_{j\sigma(AGG)}, X, S_t, \tau, \sigma_{i(AGG)}^{2*}) \\ &= \omega_0 + \sum_{j=1}^H \omega_{0j} \left\{ \frac{1}{1 + \exp(-1) \left\{ \begin{aligned} &\omega_{j0} + \omega_{jM} \left(\frac{X}{S_t}\right)^* \\ &+ \omega_{j\tau} \tau^* + \omega_{j\sigma(AGG)} (\sigma_{i(AGG)\alpha}^{2*}) \end{aligned} \right\}} \right\} \end{aligned} \quad (20)$$

Where:

- $\omega_0$  = The bias to the output neuron
- $\omega_{0j}$  = For  $j = 1, 2, \dots, H$  the weights connecting the  $j^{\text{th}}$  hidden node to the output node

The net input  $I_{j0}$  to the output neuron defined in Equation 23 above is again transformed using the sigmoid function so as to ensure positivity and that values remain in the range of 0 to 1. The resultant output from the output neuron is thus of the form:

$$f_{\alpha}(\omega_3, X, S_t, \tau, \sigma_{i(AGG)}^{2*}) = \frac{1}{1 + e^{-I_{j0}}} \quad (21)$$

where,  $\omega_3 = (\omega_0, \omega_{0j}, \omega_{j0}, \omega_{jM}, \omega_{j\tau}, \omega_{j\sigma(AGG)})$ .

Finally, a reverse of expression 13, 14 and 15 i.e.:

$$\begin{aligned} \gamma_{3\alpha}(\omega_3, X, S_t, \tau, \sigma_{i(AGG)}^{2*}) &= f_{\alpha}(\omega_3, X, S_t, \tau, \sigma_{i(AGG)}^{2*}) \\ &\left[ \begin{aligned} &\text{Max}\left\{f(\omega_3, X, S_t, \tau, \sigma_{i(AGG)}^{2*})\right\} \\ &-\text{Min}\left\{f(\omega_3, X, S_t, \tau, \sigma_{i(AGG)}^{2*})\right\} \end{aligned} \right] \\ &+ \text{Min}\left\{f(\omega_3, X, S_t, \tau, \sigma_{i(AGG)}^{2*})\right\} \end{aligned} \quad (22)$$

descales the scaled neuron output and produces the target output  $\gamma_{3\alpha}(\omega_3, X, S_t, \tau, \sigma_{i(AGG)}^{2*})$  for each  $\alpha$ , for  $\alpha = 1, 2, \dots, n$ . Equation 22 gives the target output of the model whose difference with the real valued option prices  $\hat{P}_c$  we would like to minimize in training.

### Training of the Network

The SSE is used to train face forward networks and weights are adjusted in such a way that the SSE between the targets  $\hat{P}_c$  and the model outputs, now  $f(\omega_3, X, S_t, \tau, \sigma_{i(AGG)}^{2*})$  is minimized. The SSE is defined as:

$$\begin{aligned} &S^2 \left\{ f_{\alpha}(\omega_3, X, S_t, \tau, \sigma_{i(AGG)}^{2*}) \right\} \\ &= \sum_{\alpha=1}^K \left\{ \tilde{P}_{c\alpha} - \gamma_{3\alpha}(\omega_3, X, S_t, \tau, \sigma_{i(AGG)}^{2*}) \right\}^2 \end{aligned} \quad (23)$$

where,  $\gamma_{3\alpha}(\omega_3, X, S_t, \tau, \sigma_{i(AGG)}^{2*})$  = Model output defined in Equation 23.

## Results and Discussion

In validating the developed model, the study used intraday data for the AAPL stock option for the period between December 2016 and March 2017 with 56,238 data points. Of these options values, 59.15% were in the money (ITM) while 40.85% were out of the money (OTM). The data was divided into three sets with 50% (28,119) used for training the model, 25% (14,160) used in testing and the remaining 25% (14,059) used in the validation of the model. The study highlighted that the hybrid option pricing model would be developed and its performance compared with the performance of a pure ANN option pricing model whose volatility was purely modelled using an ANN model.

### Pure ANN Model with Time-Varying Volatility

As for the pure ANN option pricing model, in addition to moneyness and time to maturity, the study used historical volatility as one of the option pricing variables. Due to the nature and size of the data, 10-day historical volatility for the AAPL stock option for the period between December 2016 and March 2017 was used. The conventional practice adopted in the pure ANN option price model is to make allowances for a large number of lags (in the case of volatility) with the belief that the ability of ANN to incorporate flexibility and redundancy creates a more robust model. This is exactly what was done in this section with volatility lags being varied beginning with lag one and increasing the number of lags until the optimal model was obtained. While dealing with volatility, attention was drawn to the effect of the lag variation on the model complexity in terms of the

structure (no of hidden layers and nodes therein, learning rate, training time, etc.) and the model performance. The study observed these variables while varying the lags beginning with lag 1. 3-D plots were used to monitor the patterns of change for every lag with respect to the number of hidden nodes, the learning rate and the SSE.

Lag 9 was the lag that produced the minimum value of the SSE (0.01895), with the number of hidden nodes  $H = 10$  and the training time  $T = 11.08s$  – an all high training time in all the lags.

The SSE, MSE and the RMSE were used to measure the performance of the pure ANN option pricing model with time-varying volatility during training, testing and validation. Table 1 summarizes the error statistics obtained in the three phases.

An analysis of the performance of the model with regards to In-The-Money (ITM) options and Out-The-Money (OTM) (Table 2) options revealed that the

model predicted the OTM option prices better than the ITM option prices.

### The Hybrid ANN Model

For the hybrid model, the mean function of the GJR-GARCH was first obtained using the ANN model by modelling the APPL stock returns the ANN model. These stock return forecasts then formed the inputs into the GJR-GARCH model resulting to the ANN-GJR-GARCH volatility model. Figure 1 shows the ANN forecasts of the AAPL returns representing the mean function of the GJR-GARCH model.

Different orders of the ANN-GJR-GARCH were fitted and the AIC, BIC, the shibata and Hannan-Quinn statistics used to select the best model. According to these statistics, the ANN-GJR-GARCH (1,1,1) was found to be optimal in fitting the AAPL stock volatility. Its statistics are provided in Tables 3 and 4.

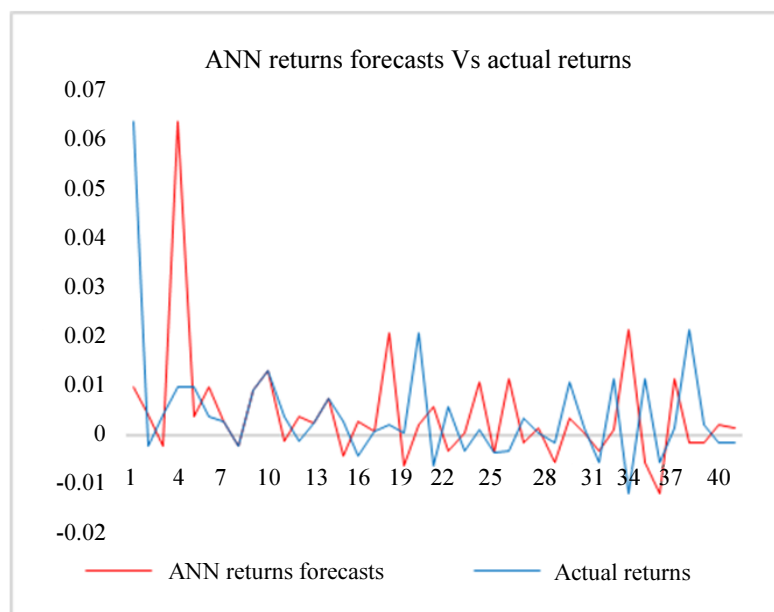


Fig. 1: ANN forecasts of the AAPL stock returns

Table 1: Performance of the pure ANN model

Performance of the pure ANN model model			
	Training (n=24,409)	Testing (n=12,204)	Validation (n=12,204)
SSE	377.51192	151.99583	324.64459
MSE	0.01546	0.01245	0.02660
RMSE	0.12436	0.11160	0.16309

Table 2: ITM-OTM Errors of the pure ANN model with non-constant volatility

	ITM (Pure ANN, n=14388)		OTM (Pure ANN, n=10026)	
	Testing	Validation	Testing	Validation
SSE	189.569	414.480	101.495	117.511
MSE	0.013	0.028	0.010	0.011
RMSE	0.114	0.169	0.100	0.108

**Table 3:** ANN-GJR-GARCH (1,1,1) Parameter estimates

	Estimate	S. Error	P-value	Nyblom S. test
Omega ( $\omega$ )	0.000000	0.000002	0.92546	0.14316
Phi ( $\Phi$ )	0.000127	0.000107	0.23792	0.06205
Theta ( $\theta$ )	0.920379	0.003133	0.00000	0.06244
Psi ( $\Psi^*$ )	-0.923005	0.015058	0.00000	0.06249
Beta( $\beta$ )	0.002707	0.000044	0.00000	0.06204

**Table 4:** ANN-GJR-GARCH (1,1,1) joint statistics

Test	AIC	BIC	Shibata	Hannan-Quinn	ARCH LM	Log-likelihood	Weighted L-Jung Box	Nyblom S. Test
Statistic	-6.346	-6.056	-6.536	-6.331	0.177	56.771	0.727	4.559

**Table 5:** Performance of the hybrid ANN model developed

Performance of the hybrid ANN model developed		Training (n=12,204)	Validation(n=12,204)
SSE		134.95529	134.34005
MSE		0.011058	0.011007
RMSE		0.105158	0.104918

**Table 6:** ITM-OTM errors of the hybrid ANN option pricing model developed

Hybrid ITM versus OTM validation errors		In-The-Money (n = 7401)	Out-of-The-Money (n = 4803)
SSE		132.85659	31.8669
MSE		0.01795	0.00663
RMSE		0.13398	0.08145

Of the 50% of the data set remaining (24, 408), 50% of this (12,204) was used in training the hybrid model while the other 50% was used for validation. The model developed consisted of 3 inputs namely: Moneyness, time to maturity and the ANN-GJR-GARCH volatility forecasts. This implies that the model developed had 3 input nodes. H was varied between 2 and 7 following the formula discussed in the methodology.

Apparently, contrary to the results obtain under the pure ANN model, where the SSE and training time varied with the number of hidden nodes, as well as the learning rate, in this case, both the SSE and the training time remained constant at 134.95529 and 0.01 respectively, the number of hidden nodes and the learning rate notwithstanding. This perhaps being an indication that the network consistently captured the same pattern between variables irrespective of the network structure. Table 5 summarizes the error statistics of the hybrid model developed.

Interest was also drawn to the performance of the model with regards to the ITM and OTM option prices. The 25% (12,204) values were categorized as either ITM or OTM in which case we had 7401 ITM cases and 4803 OTM cases. Results (as in Table 6) indicates that the hybrid ANN models performs way better in modelling OTM option prices (MSE = 0.00663) as compared to ITM option prices (MSE = 0.01795).

### Comparative Analysis of the Models Developed

One of the arguments fronted in the background of this work is that in spite of lack of a well-defined input variable selection criterion, ANN models have been developed without due consideration to the effect that the choice of an input selection technique would have on model complexity, learning difficulty and performance measures. As a result, this study sought to conduct a comparative analysis of the two neural networks models developed in line with these three attributes. As indicated in Table 7, in so far as model complexity is concerned, we start by an admission that this largely depended on the number of Input Variables (IVs). Having more IVs would definitely complicate the model structure. In our two cases, we had 9 IVs while dealing with pure ANN models and 3 in the case of the hybrid model. We note that applying an existing statistical/mathematical model in modelling an input variable using its known variables or otherwise, greatly helps in reducing the number of inputs that would otherwise have to be inputted into the neural network. Be it as obvious as it may seem, these aids a great deal in simplifying the model structure. As it can be seen, of the 2 ANN models highlighted in this paragraph, the networks eventually had 10 and 3 number of hidden nodes (H), respectively. We however note that in spite of this case, the study found that 1 layer of hidden nodes was sufficient in developing the three models, the number of inputs and the value of H notwithstanding.



**Table 7:** Comparative analysis of the models developed

Variables		Nature of the model	
Model attributes	Indicator	Pure ANN	Hybrid model
Complexity	No. of IVs	9.000000	3.000000
	Hidden nodes	10.000000	3.000000
	Hidden layers	1.000000	1.000000
Learning difficulty	Training time(s)	11.080000	0.010000
Performance (Validation)	MSE	0.026601	0.011007
	RMSE	0.163099	0.104918

In terms of learning difficulty, as depicted by the training time, the hybrid model performed better (0.01s) compared to the pure ANN model (11.08s). This can be attributed to, first, the lesser number of inputs compared to the heteroscedastic model with lags. Secondly, compared to the Pure ANN model, an existing relationship between the option prices and the other inputs, including the ANN-GJR-GARCH volatility forecasts is easily laid and captured compared to when one of the influential variables is missing. Finally, in terms of the performance of the model, it is interesting to note that the hybrid model performed better with an MSE of 0.011007, followed closely by the pure ANN model (0.026601).

## Conclusion

This study sought to deviate from modelling option prices using a pure ANN model by developing a hybrid model in which the inputs to the network are externally determined. Here, the volatility component was handled using a celebrated ANN-time series model, specifically the ANN-GJR-GARCH model - a hybrid of the ANN and a time series hybrid. The hybrid ANN option pricing model was then framed and tested with the forecasts of the ANN-GJR-GARCH model as a volatility input alongside two other inputs - time to maturity and moneyness. Finally, the study compared the performance of the hybrid model developed with that of a pure ANN model. Results indicate that all the two ANN models modelled the AAPL OTM option prices better than ITM. Secondly, the hybrid ANN model makes better predictions than the pure ANN model with time-variant volatility, with the difference between the performance of the two models being significant at 0.05 level of significance. In terms of model complexity, findings reveal that a single layer of hidden nodes remains sufficient in developing both the hybrid and the pure ANN models, the number of inputs and the value of H notwithstanding. Finally, in terms of learning difficulty, as depicted by the training time, the hybrid ANN model performed better (0.01s) compared to the pure ANN model (11.08s). This can be attributed to: First, the existence of a lesser number of inputs compared to the pure ANN model. Secondly, the ANN-

GJR-GARCH model used in modelling the volatility component best captures features that characterize financial data such as volatility clustering, leptokurtosis and leverage effects - something that ANNs in their pure forms have proved inadequate in doing.

## Acknowledgement

The authors acknowledge the African Union through the Pan African University Institute for Basic Sciences, Technology and Innovation (PAUISTI) and the department of Statistics and Actuarial Sciences (STACS) of the Jomo Kenyatta university of Agriculture and Technology for their great support.

## Funding Information

The authors acknowledge the African Union through the Pan African University Institute for Basic Sciences, Technology and Innovation (PAUISTI) for the scholarship and funding of the research leading to/and this publication.

## Author's Contributions

**Hanningtone Meshack Simiyu:** Conception and design, acquisition of data, analysis and interpretation.

**Anthony Gichuhi Waititu and Jane Aduda Akinyi:** Conception, designing and reviewing the article critically for significant intellectual content and give final approval.

## Ethics

This article is original and contains unpublished resources. All the authors have read and sanctioned the manuscript and are gratified that there are no ethical concerns involved.

## References

- Al Saedi, Y.H. and G.A. Tularam, 2018. A review of the recent advances made in the Black-Scholes models and respective solutions methods. *J. Math. Stat.*, 14: 29-39. DOI: 10.3844/jmssp.2018.29.39

- Amilon, H., 2003. A neural network versus Black-Scholes: A comparison of pricing and hedging performances. *J. Forecast.*, 22: 317-335.  
DOI: 10.1002/for.867
- Andreou, P.C., C. Charalambous and H.S. Martzoukos, 2006. Robust artificial neural networks for pricing of European options. *Comput. Econom.*, 27: 329-351.
- Ankudinova, J. and M. Ehrhardt, 2008. On the numerical solution of nonlinear Black-Scholes equations. *Comput. Math. Applic.*, 56: 799-812.  
DOI: 10.1016/j.camwa.2008.02.005
- Bohner, M. and Y. Zheng, 2009. On analytical solutions of the Black-Scholes equation. *Applied Math. Lett.*, 22: 309-313. DOI: 10.1016/j.aml.2008.04.002
- Brownlees, C.R. Engle and B. Kelly, 2011. A practical guide to volatility forecasting through calm and storm. *J. Risk*, 14: 3-22.
- Caudill, M., 1988. Neural networks primer, Part III. *AI Expert* 3.
- Černá, D., V. Pasheva, N. Popivanov and G. Venkov, 2016. Numerical solution of the Black-Scholes equation using cubic spline wavelets. *Proceedings of the AIP Conference, (IPC' 16)*, AIP Publishing, USA. DOI: 10.1063/1.4968447
- Company, R., E. Navarro, J.R. Pintos and E. Ponsoda, 2008. Numerical solution of linear and nonlinear Black-Scholes option pricing equations. *Comput. Math. Applic.*, 56: 813-821.  
DOI: 10.1016/j.camwa.2008.02.010
- Cortés, J., L. Jódar, R. Sala and P. Sevilla-Peris, 2005. Exact and numerical solution of Black-Scholes matrix equation. *Applied Math. Comput.*, 160: 607-613. DOI: 10.1016/j.amc.2003.11.018
- Edeki, S.O., O.O. Ugbebor and E.A. Owoloko, 2015. Analytical solutions of the Black-Scholes pricing model for European option valuation via a projected differential transformation method. *Entropy*, 17: 7510-7521. DOI: 10.3390/e17117510
- Engle, R.F., 1982. Autogressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflations. *Econometrica*, 50: 987-1007.
- Engle, R.F. and T. Bollerslev, 1986. Modeling the persistence of conditional variances. *Econometric Rev.*, 5: 1-50.
- Enke, D. and C.H. Dagli, 2017. A hybrid option pricing model using a neural network for estimating volatility. *Int. J. General Syst.*  
DOI: 10.1080/03081070701210303
- Forsyth, P., K. Vetzal and R. Zvan, 1999. A finite element approach to the pricing of discrete lookbacks with stochastic volatility. *Applied Math. Finance*, 6: 87-106.  
DOI: 10.1080/135048699334564
- Glosten, L., R. Jagannathan and D. Runkle, 1993. On the relationship between the expected value and the volatility of the nominal excess return on stocks. *J. Finance*, 48: 1779-1801.
- Gradojevic, N., G. Ramazan and D. Kukulj, 2007. Option pricing with modular neural networks.
- Hajela, P. and L. Berke, 1991. Neurobiological computational models in structural analysis and design. *Comput. Struct.*, 41: 657-667.  
DOI: 10.1016/0045-7949(91)90178-O
- Hajizadeha, E. and A. Seifia, 2011. A hybrid modeling approach for option pricing. *AIP Conf. Proc.*  
DOI: 10.1063/1.3663498
- Harper, J., 1994. Reducing parabolic partial differential equations to canonical form.
- Hecht-Nielsen, R., 1989. Theory of the backpropagation neural network. *Proceedings of the International Joint Conference on Neural Networks, (CNN' 89)*, IEEE Xplore Press, Washington, DC, USA, pp: 593-606.  
DOI: 10.1109/IJCNN.1989.118638
- Hull, J.S., D. Treepongkaruna, R.H. Colwell and D. Pitt, 2013. *Fundamentals of futures and options markets*. Pearson Higher Education AU.
- Hutchinson, J.M., A.W. Lo and T. Poggio, 1994. A nonparametric approach to pricing and hedging derivative securities via learning networks. *J. Finance*, 49: 851-889.
- Jódar, L., P. Sevilla-Peris, J. Cortés and R. Sala, 2005. A new direct method for solving the Black-Scholes equation. *Applied Math. Lett.*, 18: 29-32.  
DOI: 10.1016/j.aml.2002.12.016
- Lu, X., D. Que and G. Caoa, 2016. Volatility forecast based on the hybrid artificial neural network and GARCH-type models. *Proc. Comput. Sci.*, 91: 1044-1049.
- Malliaris, M. and L. Salchenberg, 1993. A neural network model for estimating option prices. *J. Applied Intell.*, 3: 193-206.
- Merton, R.C., 1990. Theory of rational option pricing. *Bell. J. Econ. Manage. Sci.*, 4: 141-183.  
DOI: 10.2307/3003143
- Mitra, S.K., 2012. An option pricing model that combines neural network approach and Black Scholes formula. *Global Journals Inc. (US)*.
- Mostafa, F. and T. Dillon, 2008. A neural network approach to option pricing. *Comput. Finance Applic.*, 41: 71-84. DOI: 10.2495/CF080081
- Najjar, Y.M., L.A. Basheer and M.N. Hajmeer, 1997. Computational neural networks for predictive microbiology: Methodology. *Int. J. Food Microbiol.*, 34: 27-49. DOI: 10.1016/S0168-1605(96)01168-3

- Nelson, D.B., 1991. Conditional heteroskedasticity in asset returns: A new approach. *Econometrica*, 59: 347-370.
- Rodrigo, M.R. and R.S. Mamon, 2006. An alternative approach to solving the Black-Scholes equation with time-varying parameters. *Applied Math. Lett.*, 19: 398-402. DOI: 10.1016/j.aml.2005.06.012
- Salchenberger, L.M., E.M. Cinar and N.A. Lash, 1992. Neural networks: A new tool for predicting thrift failures. *J. Dec. Sci.*, 23: 899-916. DOI: 10.1111/j.1540-5915.1992.tb00425.x
- Shin, B. and H. Kim, 2016. The solution of Black-Scholes terminal value problem by means of Laplace transform. *Glob. J. Pure Applied Math.*, 12: 4153-4158.
- Wilmott, P., S. Howison and J. Dewynne, 1995. *The Mathematics of Financial Derivatives: A Student Introduction*. 1st Edn., Cambridge University Press, Cambridge, ISBN-10: 0521497892, pp: 317.