

Du Plooy, Ryno; Venter, Pierre J.

Article

A comparison of artificial neural networks and bootstrap aggregating ensembles in a modern financial derivative pricing framework

Journal of Risk and Financial Management

Provided in Cooperation with:

MDPI – Multidisciplinary Digital Publishing Institute, Basel

Suggested Citation: Du Plooy, Ryno; Venter, Pierre J. (2021) : A comparison of artificial neural networks and bootstrap aggregating ensembles in a modern financial derivative pricing framework, Journal of Risk and Financial Management, ISSN 1911-8074, MDPI, Basel, Vol. 14, Iss. 6, pp. 1-18, <https://doi.org/10.3390/jrfm14060254>

This Version is available at:

<https://hdl.handle.net/10419/239670>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.



<https://creativecommons.org/licenses/by/4.0/>



Article

A Comparison of Artificial Neural Networks and Bootstrap Aggregating Ensembles in a Modern Financial Derivative Pricing Framework

Ryno du Plooy * and Pierre J. Venter

Department of Finance and Investment Management, University of Johannesburg, P.O. Box 524, Auckland Park 2006, South Africa; venter.pierre7@gmail.com

* Correspondence: duplooryrno@gmail.com

Abstract: In this paper, the pricing performances of two learning networks, namely an artificial neural network and a bootstrap aggregating ensemble network, were compared when pricing the Johannesburg Stock Exchange (JSE) Top 40 European call options in a modern option pricing framework using a constructed implied volatility surface. In addition to this, the numerical accuracy of the better performing network was compared to a Monte Carlo simulation in a separate numerical experiment. It was found that the bootstrap aggregating ensemble network outperformed the artificial neural network and produced price estimates within the error bounds of a Monte Carlo simulation when pricing derivatives in a multi-curve framework setting.

Keywords: artificial neural networks; vanilla option pricing; multi-curve framework; collateral; funding



Citation: du Plooy, Ryno, and Pierre J. Venter. 2021. A Comparison of Artificial Neural Networks and Bootstrap Aggregating Ensembles in a Modern Financial Derivative Pricing Framework. *Journal of Risk and Financial Management* 14: 254. <https://doi.org/10.3390/jrfm14060254>

Academic Editor: Jakub Horak

Received: 5 May 2021

Accepted: 17 May 2021

Published: 7 June 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Black and Scholes (1973) established the foundation for modern option pricing theory by showing that under certain ideal market conditions, it is possible to derive an analytically tractable solution for the price of a financial derivative. Industry practitioners however quickly discovered that certain assumptions underlying the Black–Scholes (BS) model such as constant volatility and the existence of a unique risk-free interest rate were fundamentally flawed. Following the 2007 credit crisis, Piterbarg (2010) extended the BS partial differential equation (PDE) to account for collateral and funding costs by introducing three deterministic interest rates, more specifically the funding rate r_F , the collateral rate r_C , and the repurchase agreement rate r_R^S . Analytically tractable solutions to the Piterbarg PDE were subsequently derived by Von Boetticher (2017) for European-style derivatives.

The applications of machine learning in quantitative finance have been extensively studied over the past few decades, with a large number of researchers contributing to various areas in the financial machine learning literature. According to a detailed literature review by Ruf and Wang (2020), there have been more than 100 papers published that have studied the applications of machine learning in the pricing and hedging of financial derivative securities.

Pioneering research by Hutchinson et al. (1994), Garcia and Gençay (2000), and Bennell and Sutcliffe (2004) put forth the idea that machine learning techniques such as artificial neural networks (ANNs) can be used as alternative non-parametric methods to traditional derivative pricing models. Advances in technology and the increase in machine learning resources available today have enabled researchers and practitioners to develop and implement more complex ANNs. Recent papers by Culkin and Das (2017) and Liu et al. (2019) showed that ANNs that are both complex and efficient can easily be trained and applied to financial derivative pricing problems.

Since it is well known from the existing literature that machine learning techniques are able to approximate the BS model, the question thus arises: Is it possible for these techniques to learn the closed-form solutions derived by [Von Boetticher \(2017\)](#) and be used to price derivatives in a multi-curve framework setting? Furthermore, does the use of ensemble methods such as bootstrap aggregation (bagging) as investigated in an alternative study on default risk analysis by [Hamori et al. \(2018\)](#) outperform an ANN?

The aim of this paper was to provide the answer to these questions by developing two learning networks, namely an ANN and a bagging ensemble, and comparing their performance in pricing European call options in the Piterbarg framework using option price data from the South African market.

This paper is structured as follows: Section 2 consists of the research methodology, which includes exploring the mathematical theory underpinning the Piterbarg framework, as well as the mechanics behind ANNs and ensemble methods. Section 3 entails the data generation process and configuration of the respective learning networks. Section 4 is comprised of the numerical results, and ultimately, the findings and concluding remarks are presented in Section 5.

2. Methodology

In this section, the theory applied in this paper is outlined and divided into two parts. The first part focuses on the Piterbarg framework, whereas the second part explores the mechanics behind ANNs and ensemble methods.

2.1. The Piterbarg Framework

The Piterbarg framework is a simple extension of the BS model that incorporates collateral to offset liabilities that arise in the event of default. To better understand the impact of collateral on the price of a derivative, it is first useful to understand how collateral is connected to the credit riskiness associated with a certain contract or agreement. If a party posts cash when entering into an agreement, the collateral rate r_C is used to calculate the interest payments, and since cash is considered a less risky asset, the collateral rate tends to be low. If a party enters into a repurchase agreement, which is simply a collateralized loan, then the party that borrows cash will need to later repay the amount borrowed together with interest at a prespecified date. In addition to this, the party will also put up an asset as collateral. Since the asset posted as collateral can be either a bond where the issuer can default or a share whose value can fluctuate, there is more credit risk associated with this agreement. This implies that the repurchase agreement rate r_R^S should be higher than the collateral rate. If a party enters into an agreement and posts no collateral, the loan is unsecured, which exposes the lender to more credit risk. This implies a higher interest rate, with the funding rate r_F being the highest of the three interest rates. Taking the connection between the different interest rates into account, the following relation must hold:

$$r_C \leq r_R^S \leq r_F. \quad (1)$$

According to [Hunzinger and Labuschagne \(2015\)](#), if a derivative is fully collateralized, the expected payoff of the derivative is discounted using the collateral curve, and if no collateral is posted, then the expected payoff is discounted using the funding curve. Therefore, given the relation between the three deterministic interest rates in Equation (1), it follows that:

$$V_{FC}(S_t, t) \geq V_{ZC}(S_t, t), \quad (2)$$

where the fully collateralized derivative $V_{FC}(S_t, t)$ discounted at the collateral rate is more expensive than the corresponding zero collateral derivative $V_{ZC}(S_t, t)$ discounted at the funding rate. The derivation of the Piterbarg partial differential equation (PDE) is presented in the next section.

2.2. The Piterbarg PDE

In this section, it is shown how the PDE for a general collateralized derivative can be derived by constructing a self-financing replicating portfolio. Piterbarg (2010) assumed the process of the underlying asset S_t at time t under the risk-neutral measure $\mathbb{Q}_{r_R^S}$ is given by:

$$dS_t = r_{R_t}^S S_t dt + \sigma_P S_t dW_t, \tag{3}$$

where the drift term is the repurchase rate $r_{R_t}^S$ with respect to the underlying asset, σ_P is the volatility of the underlying asset, and W_t is a standard Brownian motion under the measure $\mathbb{Q}_{r_R^S}$. The change in the value of the derivative security V_t is obtained through the application of Itô's lemma and is given by:

$$dV_t = \left[\frac{\partial V_t}{\partial t} + a \frac{\partial V_t}{\partial S_t} + \frac{1}{2} b^2 \frac{\partial^2 V_t}{\partial S_t^2} \right] dt + b \frac{\partial V_t}{\partial S_t} dW_t, \tag{4}$$

where $a = r_{R_t}^S S_t$ and $b = \sigma_P S_t$. Let $V_t = \Pi_t$ denote the value of a replicating portfolio, where Δ_t units of the underlying asset and γ_t units of cash are held. More formally, this is represented as:

$$\Pi_t = \Delta_t S_t + \gamma_t. \tag{5}$$

The cash amount γ_t can be invested in or borrowed from various accounts, namely the funding account γ_{F_t} , the collateral account γ_{C_t} , and the repurchase account γ_{R_t} , which yields the following replicating portfolio:

$$\Pi_t = \Delta_t S_t + \gamma_{F_t} + \gamma_{C_t} - \gamma_{R_t}. \tag{6}$$

It was assumed that the funding account accrues at the funding rate r_{F_t} , the collateral account accrues at the collateral rate r_{C_t} , and the repurchase account accrues at the repurchase agreement rate $r_{R_t}^S$. The replicating portfolio is considered to be self-financing if changes in the value of the portfolio are only due to changes in the value of the assets held in the portfolio. Thus, from the self-financing condition, the change in the value of the portfolio at time t is given by:

$$d\Pi_t = \Delta_t dS_t + \left(\gamma_{F_t} r_{F_t} + \gamma_{C_t} r_{C_t} - \gamma_{R_t} r_{R_t}^S \right) dt. \tag{7}$$

Using the results from Equations (3) and (4) and letting $\Delta_t = \frac{\partial V_t}{\partial S_t}$ to eliminate the risk in the portfolio due to changes in the underlying asset result in:

$$\frac{\partial V_t}{\partial t} dt + \frac{1}{2} \sigma_P^2 S_t^2 \frac{\partial^2 V_t}{\partial S_t^2} dt = \left(\gamma_{F_t} r_{F_t} + \gamma_{C_t} r_{C_t} - \gamma_{R_t} r_{R_t}^S \right) dt. \tag{8}$$

To ensure the absence of any arbitrage opportunities, it is required that $\gamma_{F_t} = V_t - \gamma_{C_t}$, $\gamma_{C_t} = C_t$, where C_t is the value of collateral posted and $\gamma_{R_t} = \Delta_t S_t$. Incorporating this into Equation (8) results in the Piterbarg PDE:

$$\frac{\partial V_t}{\partial t} + r_{R_t}^S S_t \frac{\partial V_t}{\partial S_t} + \frac{1}{2} \sigma_P^2 S_t^2 \frac{\partial^2 V_t}{\partial S_t^2} = V_t r_{F_t} - C_t (r_{F_t} - r_{C_t}). \tag{9}$$

The closed-form solutions for zero collateral and fully collateralized European call options are given below.

2.3. Solutions for European Call Options in the Piterbarg Framework

Since it was shown that the value of a derivative that incorporates collateral satisfies the PDE in Equation (9), the closed-form solutions for zero collateral (ZC) and fully collat-

eralized (FC) European call options based on the derivations by Von Boetticher (2017) are given by:

$$V_{ZC}(S_t, t) = e^{-\int_t^T r_F(u)du} \left[S_t e^{\int_t^T r_R^S(u)du} \Phi(d_1) - K \Phi(d_2) \right], \tag{10}$$

and:

$$V_{FC}(S_t, t) = e^{-\int_t^T r_C(u)du} \left[S_t e^{\int_t^T r_R^S(u)du} \Phi(d_1) - K \Phi(d_2) \right], \tag{11}$$

where:

$$d_1 = \frac{\ln\left(\frac{S_t}{K}\right) + \int_t^T r_R^S(u)du + \frac{1}{2}\sigma_P^2\tau}{\sigma_P\sqrt{\tau}},$$

$$d_2 = d_1 - \sigma_P\sqrt{\tau}, \tag{12}$$

and where τ is the time-to-maturity (year fraction). The theory behind ANNs and ensemble methods that form the backbone of this paper will be considered in the next section.

2.4. Neural Network Modeling Approaches

This section briefly provides some background on the mechanics behind ANNs, as well as shows how ensemble methods are able to reduce the variance of the predictions of individual ANNs.

2.4.1. Artificial Neural Networks

ANNs were initially inspired by neuroscience and intended to imitate the actions of the biological brain. In modern times, ANNs can be more formally defined as function approximation machines designed to achieve statistical generalization (Goodfellow et al. 2016). The general structure of a feed-forward ANN is illustrated in Figure 1.

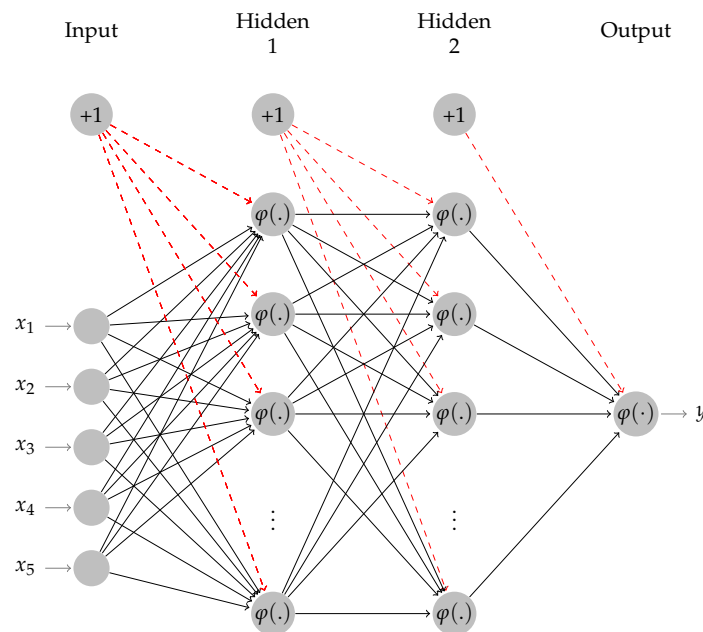


Figure 1. Feed-forward artificial neural network structure.

Where in Figure 1, x represents the inputs, the interconnected lines represent the synaptic weights, $+1$ represents the bias term, $\varphi(\cdot)$ represents the activation function, and y is the output of the ANN. Once the flow of information as illustrated in Figure 1 has passed through the network, the error between the actual and predicted outcome was

calculated, and a correction to the synaptic weights throughout the network was applied using a back-propagation algorithm with the aim of minimizing this error.

2.4.2. Ensemble Methods

Ensemble methods are variance reduction techniques that can be used to improve the accuracy of predictions since a single learner or ANN is often plagued by a multitude of estimation problems. In essence, an ensemble network consists of a finite set N of weak learners that are combined to create a superior learner that will theoretically outperform any of the individual learners. A common ensemble method is bootstrap aggregating (bagging) predictors by Breiman (1996), where N subsets of training sets are randomly sampled with replacement to train each member i where $i = 1, \dots, N$. After fitting each member on a respective training set, a bagging ensemble is formed by combining the predictions of each member and taking the average of the individual predictions to form an ensemble prediction. As highlighted by De Prado (2018), a bagging ensemble will reduce the variance of predictions if and only if the individual members are not perfectly correlated. More formally, the magnitude of the variance reduction achieved by a bagging ensemble can be illustrated using an intuitive proof by Bishop (2006). Suppose that the output of each individual member, or ANN, can be written as a function of the actual value $h(\mathbf{x})$ and some error term $\epsilon_i(\mathbf{x})$ and is given by:

$$y_i(\mathbf{x}) = h(\mathbf{x}) + \epsilon_i(\mathbf{x})$$

and the average sum-of-squared error is given by:

$$\mathbb{E}_{\mathbf{x}} \left[(y_i(\mathbf{x}) - h(\mathbf{x}))^2 \right] = \mathbb{E}_{\mathbf{x}} \left[\epsilon_i(\mathbf{x})^2 \right],$$

where $\mathbb{E}_{\mathbf{x}}[\cdot]$ is the expectation with respect to the distribution of vector \mathbf{x} . Thus, the average error of each ANN in the bagging ensemble can be written as:

$$E_{ANN} = \frac{1}{N} \sum_{i=1}^N \mathbb{E}_{\mathbf{x}} \left[\epsilon_i(\mathbf{x})^2 \right].$$

If the prediction of the bagging ensemble is given by:

$$y_{Ensemble}(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^N y_i(\mathbf{x}),$$

then the expected error from the bagging ensemble can be written as:

$$\begin{aligned} E_{Ensemble} &= \mathbb{E}_{\mathbf{x}} \left[\left(\frac{1}{N} \sum_{i=1}^N \epsilon_i(\mathbf{x}) \right)^2 \right] \\ &= \frac{1}{N^2} \mathbb{E}_{\mathbf{x}} \left[\sum_{i=1}^N \sum_{j=1}^N \epsilon_i(\mathbf{x}) \epsilon_j(\mathbf{x}) \right] \\ &= \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N \mathbb{E}_{\mathbf{x}} \left[\epsilon_i(\mathbf{x}) \epsilon_j(\mathbf{x}) \right]. \end{aligned} \tag{13}$$

If we assume that the errors have a zero mean and are uncorrelated such that the following holds:

$$\begin{aligned} \mathbb{E}_{\mathbf{x}}[\epsilon_i(\mathbf{x})] &= 0, \\ \mathbb{E}_{\mathbf{x}}[\epsilon_i(\mathbf{x})\epsilon_j(\mathbf{x})] &= 0, \quad \text{for } i \neq j \end{aligned}$$

then Equation (13) reduces to:

$$\begin{aligned}
 E_{Ensemble} &= \frac{1}{N^2} \sum_{i=1}^N \mathbb{E}_{\mathbf{x}} \left[\epsilon_i(\mathbf{x})^2 \right] \\
 &= \frac{1}{N} E_{ANN}.
 \end{aligned}
 \tag{14}$$

From the result in Equation (14), the expected error of a bagging ensemble can be reduced increasing the number of N copies of a network given that the errors of the individual ANNs are uncorrelated. The next section explores the data generation process, as well as the configuration of the base learners.

3. Data Generation and Base Learner Configuration

The following section outlines the steps followed to generate artificial option price data for training the two learning networks. The process of transforming the constructed implied volatility surface into an option price surface to be used as the testing set and the optimal configuration of the base learners is also discussed.

3.1. Training Data

The availability of option price data is largely scarce in the South African market due to illiquidity; therefore, it was required to generate artificial training data using a similar approach to that of [Culkin and Das \(2017\)](#) and [Liu et al. \(2019\)](#). The artificial data were generated by randomly sampling input data from a wide range of parameter values, which were converted to zero collateral and fully collateralized European call option prices using the respective closed-form solutions. Under the assumption of constant interest rates, the closed-form solutions in Equations (10) and (11) according to [Levendis and Venter \(2019\)](#) can be simplified and rewritten as:

$$\begin{aligned}
 V_{ZC}(S_t, t) &= e^{-\int_t^T r_F(u)du} \left[S_t e^{\int_t^T r_R^S(u)du} \Phi(d_1) - K \Phi(d_2) \right] \\
 &= e^{-r_F \tau} \left[S_t e^{r_R^S \tau} \Phi(d_1) - K \Phi(d_2) \right],
 \end{aligned}
 \tag{15}$$

and:

$$\begin{aligned}
 V_{FC}(S_t, t) &= e^{-\int_t^T r_C(u)du} \left[S_t e^{\int_t^T r_R^S(u)du} \Phi(d_1) - K \Phi(d_2) \right] \\
 &= e^{-r_C \tau} \left[S_t e^{r_R^S \tau} \Phi(d_1) - K \Phi(d_2) \right],
 \end{aligned}
 \tag{16}$$

where:

$$\begin{aligned}
 d_1 &= \frac{\ln\left(\frac{S_t}{K}\right) + \left(r_R^S + \frac{1}{2}\sigma_P^2\right)\tau}{\sigma_P \sqrt{\tau}}, \\
 d_2 &= d_1 - \sigma_P \sqrt{\tau}.
 \end{aligned}$$

The parameter ranges used to generate artificial price data are outlined in Table 1.

Table 1. Training set input parameter ranges.

| Parameter | Range |
|---|----------------------------------|
| Spot price of the underlying asset (S_t) ¹ | (R10,000, R150,000) |
| Strike price (K) | (60.00% to 140.00% of S_t) |
| Time-to-maturity (τ) | (7/365, 3.5) |
| Repurchase agreement rate (r_R^S) | (3.00%, 35.00%) |
| Collateral rate (r_C) | (60.00% to 80.00% of r_R^S) |
| Funding rate (r_F) | (120.00% to 140.00% of r_R^S) |
| Implied volatility (σ_P) | (2.00%, 80.00%) |

¹ Prices are denoted in Rand (R), which is the domestic currency for South African markets.

The number of samples randomly generated from the input parameter ranges in Table 1 for training and validating the two learning networks, as well as the general specifications are outlined in Table 2.

Table 2. Training and validation set specifications.

| Configuration | ANN | Bagging Ensemble |
|---------------------------|-----------|------------------|
| Training Sample Size | 1,500,000 | 1,500,000 |
| Number of Members | 1 | 25 |
| Sampling with Replacement | N/A | Yes |
| Training Split | 85% | 85% |
| Validation Split | 15% | 15% |

As outlined in Table 2, the bagging ensemble was constructed by sampling data with replacement from the original training sample to produce 25 new training sets of equal size to the original training sample.

The features of the input and output training sets were scaled using the homogeneity hint by Merton (1973) in a similar fashion to that of Hutchinson et al. (1994), Garcia and Gençay (2000), Bennell and Sutcliffe (2004), Culkin and Das (2017), and Liu et al. (2019). In essence, the homogeneity hint involves scaling the spot price of the underlying asset and the option price with the strike price, which resulted in the following input training sets:

$$\begin{aligned}
 \mathbf{x}_{ZC:Train} &= \{S_t/K, \tau, r_R^S, r_F, \sigma_P\}, \\
 \mathbf{x}_{FC:Train} &= \{S_t/K, \tau, r_R^S, r_C, \sigma_P\},
 \end{aligned}$$

and outputs of the form:

$$\begin{aligned}
 \mathbf{y}_{ZC:Train} &= \{V_{ZC}(S_t, t)/K\}, \\
 \mathbf{y}_{FC:Train} &= \{V_{FC}(S_t, t)/K\}.
 \end{aligned}$$

The construction of the testing set is discussed in the next section.

3.2. Testing Data

The testing set for pricing zero collateral and fully collateralized European call options consisted of the JSE Top 40 European call option price data obtained from the JSE dated 9 April 2019. From the option strike prices and implied volatility parameters, an implied volatility surface was constructed using linear interpolation in the strike and linear variance in the maturity space, which was consistent with market best practices. This resulted in a surface consisting of 10,000 implied volatility estimates. Since there are no published zero collateral or fully collateralized European call option price data in the South African market and the implied volatility surface was based on the BS model, the following simplifications had to be made:

1. If zero collateral trades were considered, it was assumed that the implied volatility surface was constructed using zero collateral trades;
2. If fully collateralized trades were considered, it was assumed that the implied volatility surface was constructed using fully collateralized trades.

The price of the underlying index (JSE Top 40) on the valuation date was R51,564.09, and it was assumed that $r_C = 5.50\%$, $r_R^S = 7.00\%$, $r_F = 8.50\%$ and that the continuous dividend yield was equal to zero. Since the rates were assumed to be constant, the closed-form solutions in Equations (15) and (16) were used to convert the implied volatility surface into zero collateral and fully collateralized European call option price surfaces. The constructed implied volatility surfaces and price surfaces are shown in Figures 2 and 3.

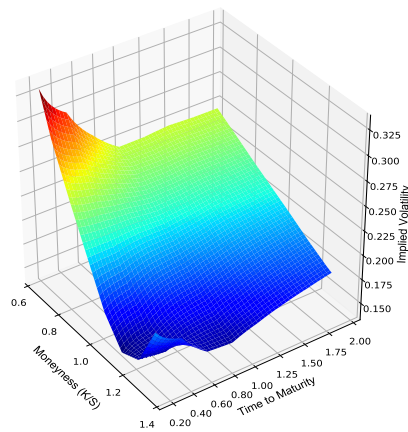


Figure 2. Implied volatility surface.

(a) Zero collateral price surface

(b) Fully collateralized price surface

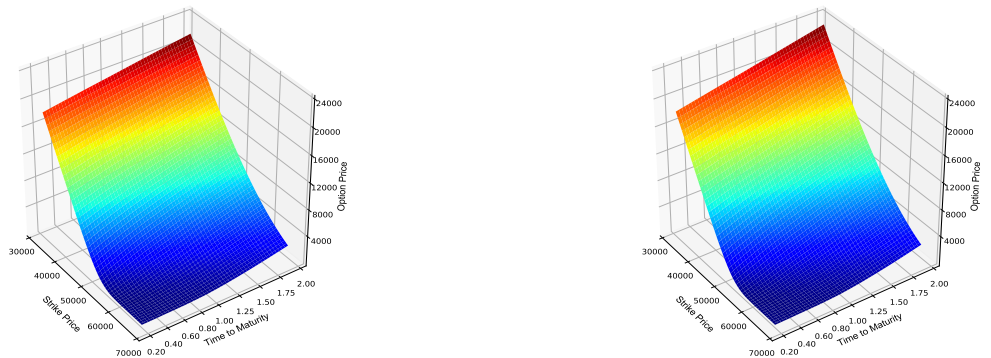


Figure 3. Price surfaces.

As seen in Figure 3, especially for deep “out-of-the-money” and long-dated trades, the fully collateralized trades discounted at the collateral rate were more expensive than the corresponding zero collateral trades discounted at the funding rate. The input testing sets are given by:

$$x_{ZC:Test} = \{S_t/K, \tau, r_R^S, r_F, \sigma_P\},$$

$$x_{FC:Test} = \{S_t/K, \tau, r_R^S, r_C, \sigma_P\},$$

and the outputs to evaluate the two learning networks are given by:

$$y_{ZC:Test} = \{V_{ZC}(S_t, t)/K\},$$

$$y_{FC:Test} = \{V_{FC}(S_t, t)/K\}.$$

The configuration of the base learner is presented in the next section.

3.3. Base Learner Configuration

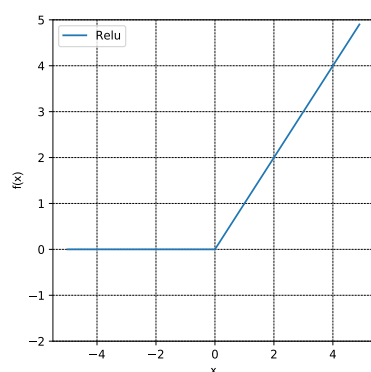
In this section, the process involved in establishing the optimal ANN configuration is presented. Several different combinations of hyperparameters such as the number of layers, neurons per layers, and batch sizes were drafted and evaluated on a smaller data set, which consisted of 150,000 samples generated from the input parameter ranges using a grid-search cross-validation (CV) technique. The collective best ANN configuration for both zero collateral and fully collateralized European call options is reported in Table 3.

Table 3. ANN configuration.

| Parameter | Configuration |
|----------------------------------|---------------|
| Number of hidden layers | 2 |
| Neurons in first hidden layer | 512 |
| Neurons in second hidden layer | 512 |
| Neurons in output layer | 1 |
| Hidden layer activation function | ReLU |
| Output layer activation function | Softplus |
| Optimizer | Adam |
| Batch size | 64 |
| Epochs | 20 |

The configurations reported in Table 3 outlines the structure of the ANN, as well as that of each member that forms part of the bagging ensemble. It is important to note that this exercise was solely performed to get a better indication of what the optimal ANN configuration might be instead of using a “trial-and-error” approach. According to [Cybenko \(1989\)](#) and [Hornik et al. \(1989\)](#), a feed-forward ANN with a single hidden layer and a continuous non-linear activation function is able to approximate any continuous function; thus, the proposed ANN configuration will be satisfactory. The chosen activation functions included the rectified linear unit (ReLU) function for all hidden layers and the Softplus function for the output layer. A key factor for using the Softplus function in the output layer is to ensure that the price predictions are non-negative, since the value of a call option cannot be negative. The graphical illustrations of the ReLU and Softplus functions can be seen in Figure 4.

(a) Rectified linear units (ReLUs)



(b) Softplus

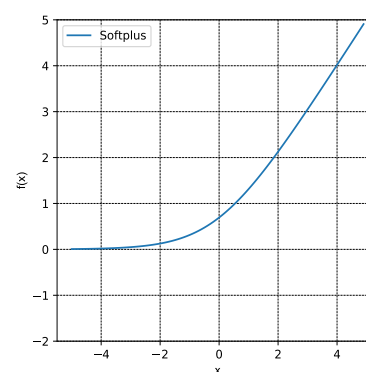


Figure 4. Activation functions.

Other configurations included the use of the adaptive moment estimation (Adam) optimizer by [Kingma and Ba \(2014\)](#) and the use of early-stopping checkpoints during the training of each ANN, which monitored validation losses to prevent overfitting the ANN. The results of this paper are presented in the next section.

4. Results

This section presents the numerical results of deploying the two learning networks, namely an ANN and a bagging ensemble to price zero collateral and fully collateralized European call options using a constructed implied volatility surface. This section also compares the pricing performance of a bagging ensemble to a Monte Carlo simulation in a separate numerical experiment. The two learning networks were developed and implemented in Python using the Keras Application Programming Interface (API) based on TensorFlow 2.0 developed by Chollet (2015). All calculations were performed on a Dell Inspiron 3567 i5-7200U CPU @ 2.50GHz with 4GB of installed RAM.

4.1. Zero Collateral Numerical Results

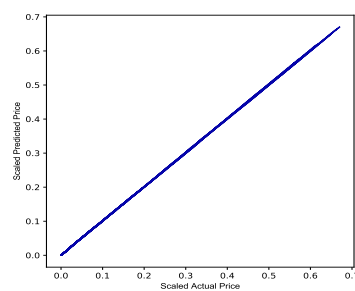
After training the two learning networks on the zero collateral training set and storing the optimal configurations for each network, the testing set input parameters were given to the two trained learning networks, which resulted in predicted prices of the form: $V_{ZC}(S_t, t)/K$. The performance of the two learning networks on the zero collateral testing set is reported in Table 4.

Table 4. Performance metrics: zero collateral.

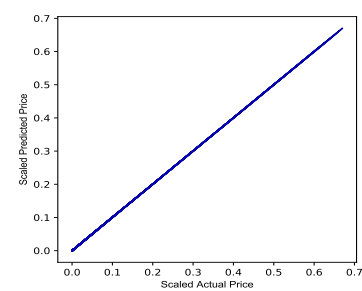
| Network Type | MSE | RMSE | R^2 |
|------------------|-----------------------|----------|----------|
| ANN | 1.67×10^{-6} | 0.001291 | 0.999948 |
| Bagging Ensemble | 1.23×10^{-7} | 0.000350 | 0.999996 |

From Table 4, it is clear that both learning networks produced accurate predictions when compared to the actual prices obtained using the closed-form solution. Based on the mean-squared-error (MSE) and root-mean-squared-error (RMSE), the use of a 25-member bagging ensemble resulted in a substantial improvement in accuracy over the ANN. The coefficient of determination (R^2) reported for the two learning networks was indifferent. A graphical representation of the performance of the two learning networks is illustrated in Figure 5.

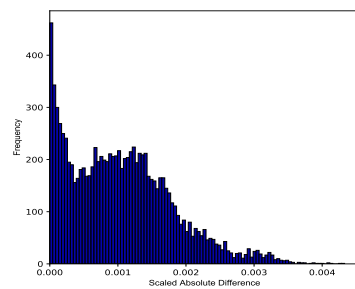
(a) Actual vs. Predicted: ANN



(b) Actual vs. Predicted: Bagging Ensemble



(c) Absolute Error: ANN



(d) Absolute Error: Bagging Ensemble

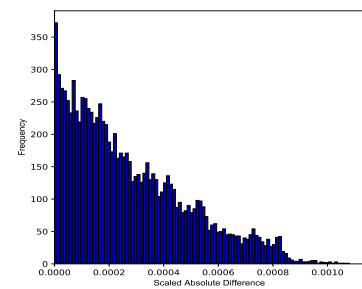
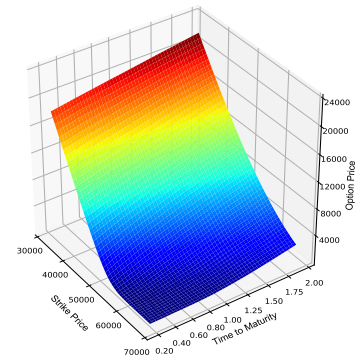
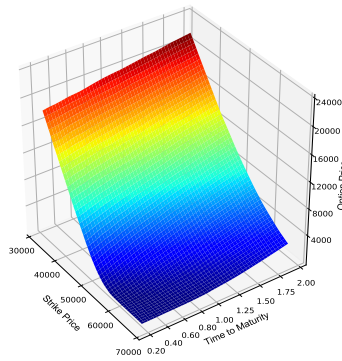


Figure 5. Actual vs. predicted results: zero collateral.

The prices predicted by the two learning networks were converted back into a zero collateral European call option price surface and compared to the actual price surface in Figure 3a. The results are shown in Figure 6.

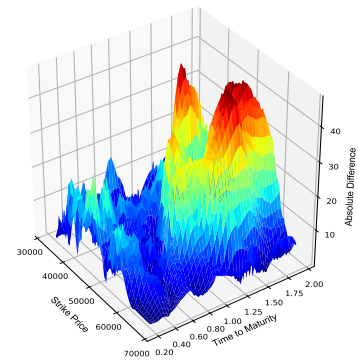
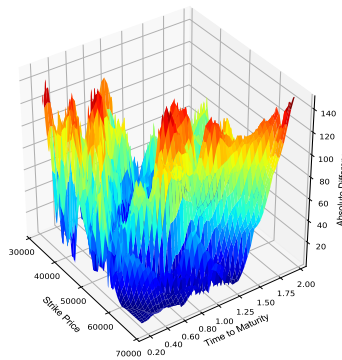
(a) Price surface: ANN

(b) Price surface: Bagging Ensemble



(c) Absolute error: ANN

(d) Absolute error: Bagging Ensemble



(e) Relative error: ANN

(f) Relative error: Bagging Ensemble

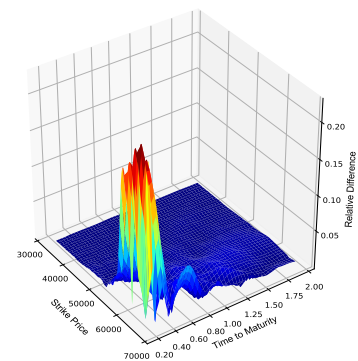
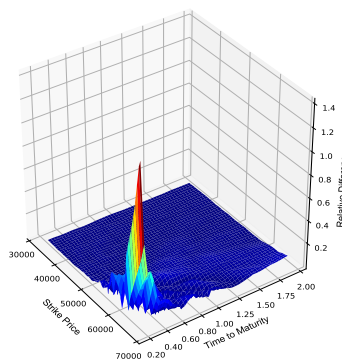


Figure 6. JSE Top 40 zero collateral call options.

As illustrated in Figure 6a,b, it is clear that both learning networks generated price surfaces graphically identical to the original price surfaces. The evaluation of the absolute errors in Figure 6c,d shows that the absolute error was reduced substantially across the surface when using a bagging ensemble. This is also evident when comparing the relative errors in Figure 6e,f. The peak in the relative error for deep “out-of-the-money” options was attributable to the Softplus activation function used in the output layer, which resulted in a slight overestimation bias since the function was not zero-centered. A more detailed view of the magnitude of errors is reported in Table 5.

Table 5. Magnitude of errors: zero collateral.

| Metric | Analytical | ANN | Bagging Ensemble |
|--------------------|------------|------------|------------------|
| Min price | R1.32 | R3.17 | R1.61 |
| Max price | R22,219.81 | R22,221.17 | R22,216.90 |
| Min absolute error | N/A | R0.00 | R0.00 |
| Max absolute error | N/A | R148.30 | R47.62 |

From the values reported in Table 5, the use of a bagging ensemble resulted in a decrease of roughly 68% in the maximum absolute error over the ANN. The minimum price generated by the bagging ensemble was also more representative of the actual minimum price obtained using the closed-form solution. The application of the two learning networks in the pricing of fully collateralized European call options will be discussed in the next section.

4.2. Fully Collateralized Numerical Results

The same procedure as mentioned earlier for the zero collateral case was followed to train and store the optimal configurations of the two learning networks applied to fully collateralized trades. The fully collateralized testing set was given to the two trained learning networks, which resulted in predicted prices of the form: $V_{FC}(S_t, t)/K$. The performance of the two learning networks on the fully collateralized testing set is reported in Table 6.

Table 6. Performance metrics: fully collateralized.

| Network Type | MSE | RMSE | R^2 |
|------------------|-----------------------|----------|----------|
| ANN | 1.63×10^{-6} | 0.001276 | 0.999952 |
| Bagging Ensemble | 1.99×10^{-7} | 0.000446 | 0.999994 |

It is clear from Table 6 that the bagging ensemble once again produced more favorable results based on the MSE, RMSE, and R^2 metrics compared to the ANN. A graphical representation of the performance of the two learning networks on the fully collateralized testing set is shown in Figure 7.

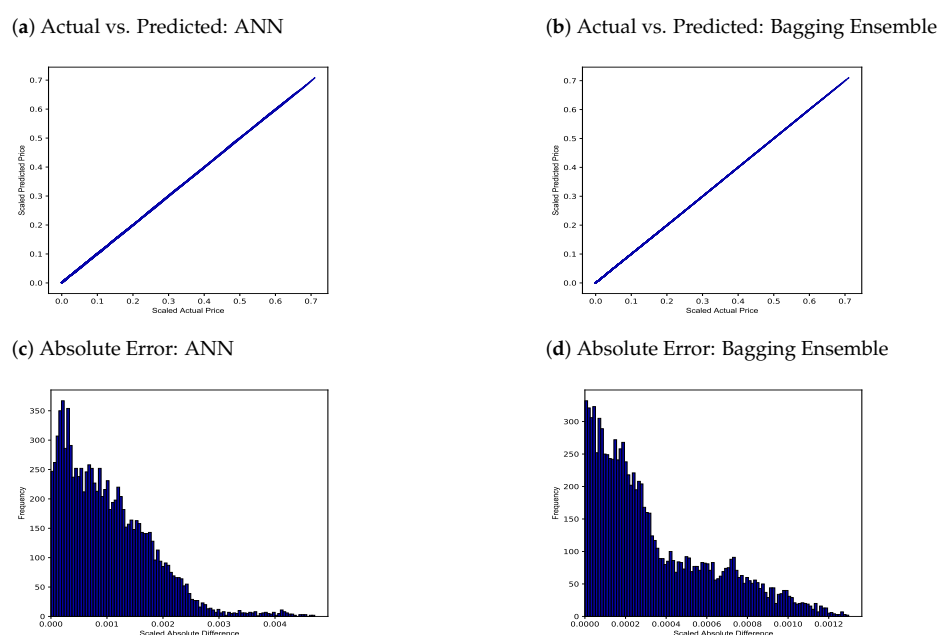
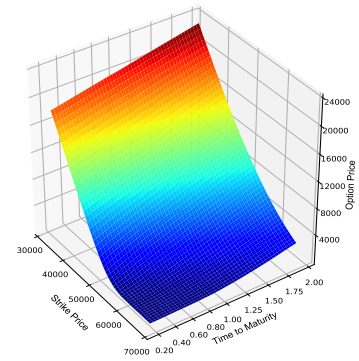
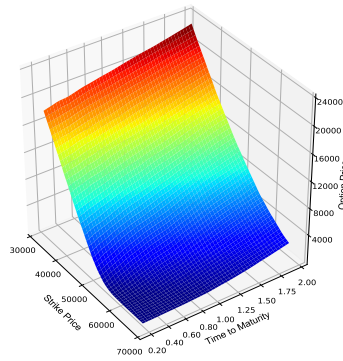


Figure 7. Actual vs. predicted results: fully collateralized.

The predictions of the two learning networks were converted back into a fully collateralized European call option price surface and compared against the actual price surface in Figure 3b. The results are illustrated in Figure 8.

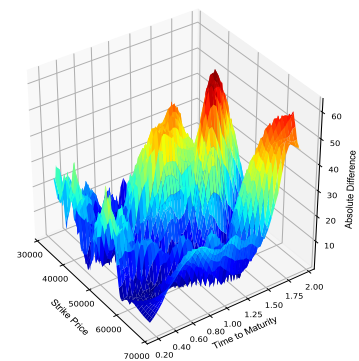
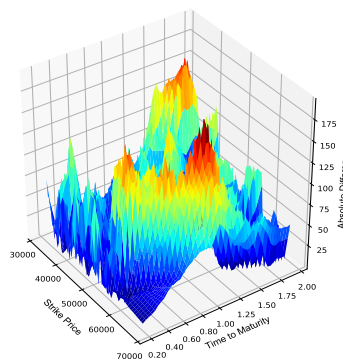
(a) Price surface: ANN

(b) Price surface: Bagging Ensemble



(c) Absolute error: ANN

(d) Absolute error: Bagging Ensemble



(e) Relative error: ANN

(f) Relative error: Bagging Ensemble

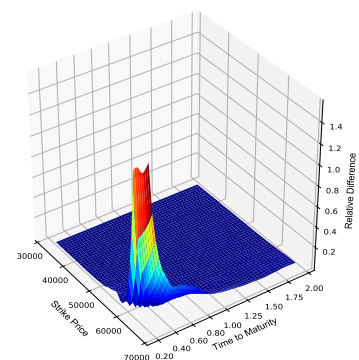
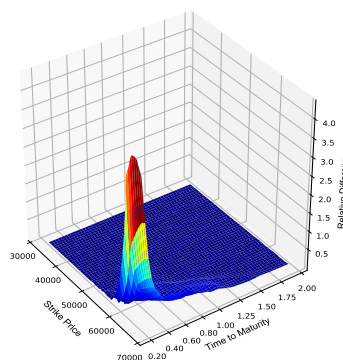


Figure 8. JSE Top 40 fully collateralized call options.

From Figure 8, the same observations can be made as in Figure 6, and it is clear that the bagging ensemble outperformed the ANN. A more detailed view of the improvement in performance can be seen in Table 7.

Table 7. Magnitude of errors: fully collateralized.

| Metric | Analytical | ANN | Bagging Ensemble |
|--------------------|------------|------------|------------------|
| Min price | R1.32 | R6.65 | R3.40 |
| Max price | R23,553.12 | R23,479.49 | R23,543.68 |
| Min absolute error | N/A | R0.02 | R0.00 |
| Max absolute error | N/A | R194.81 | R63.68 |

The values reported in Table 7 show that the use of a bagging ensemble resulted in a roughly 67% decrease in the maximum absolute error, as well as a more representative minimum price compared to the ANN. It is, thus, evident that the use of a bagging ensemble consistently produced more accurate and realistic predictions when pricing zero collateral and fully collateralized European call options. To get a better grasp on the pricing capabilities of the bagging ensemble, the next section will compare the performance of the bagging ensemble to one of the most widely used numerical techniques, namely a Monte Carlo simulation.

4.3. Numerical Results: Bagging Ensemble vs. Monte Carlo Simulation

The aim of this section is to illustrate the potential that can be unlocked by exploring alternative numerical methods such as a bagging ensemble in the case of this paper. For the outcome of this numerical experiment to be of any significance, it is first necessary to state the following:

1. The stochastic process of the underlying asset follows a geometric Brownian motion;
2. Constant interest rates were assumed;
3. The underlying asset does not pay any dividends;
4. Trades are European in nature;
5. Trades are devoid of any friction costs;
6. An Actual/365 day-count convention is used;
7. The implied volatility parameters obtained from the volatility skew dated 9 April 2019 were assumed to be constructed using either zero collateral or a fully collateralized trades, depending on which type of trade was considered;
8. We assumed vanilla options only.

Furthermore, the following two critical conditions must also hold:

1. Given the relation in Equation (2), the price of a zero collateral European call option must be less than that of a fully collateralized European call option for any trade;
2. For the bagging ensemble to be a viable alternative to a Monte Carlo simulation, it must be shown that the numerical accuracy of the bagging ensemble is within the three standard deviation error bounds of Monte Carlo simulation estimates for a reasonable number of simulations.

The zero collateral and fully collateralized trade information that served as the basis for this numerical experiment are outlined in Tables 8 and 9.

Table 8. Zero collateral trade specifications.

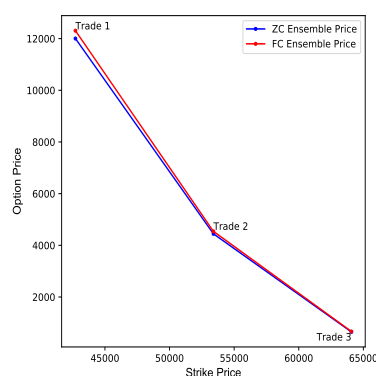
| Parameter | Trade 1 | Trade 2 | Trade 3 |
|--|------------|------------|------------|
| Trade type | Call | Call | Call |
| Spot price of the underlying asset (S_t) | R51,564.09 | R51,564.09 | R51,564.09 |
| Strike (K) | R42,716.80 | R53,396.00 | R64,075.20 |
| Time to maturity | 345 days | 345 days | 345 days |
| Implied volatility (σ_P) | 22.32% | 18.50% | 15.17% |
| Repurchase agreement rate (r_R^S) | 7.00% | 7.00% | 7.00% |
| Funding rate (r_F) | 8.50% | 8.50% | 8.50% |

Table 9. Fully collateralized trade specifications.

| Parameter | Trade 1 | Trade 2 | Trade 3 |
|--|------------|------------|------------|
| Trade type | Call | Call | Call |
| Spot price of the underlying asset (S_t) | R51,564.09 | R51,564.09 | R51,564.09 |
| Strike (K) | R42,716.80 | R53,396.00 | R64,075.20 |
| Time to maturity | 345 days | 345 days | 345 days |
| Implied volatility (σ_P) | 22.32% | 18.50% | 15.17% |
| Repurchase agreement rate (r_R^S) | 7.00% | 7.00% | 7.00% |
| Collateral rate (r_C) | 5.50% | 5.50% | 5.50% |

From the trade information in Tables 8 and 9, the prices for the respective trades generated by the bagging ensemble can be seen in Figure 9.

(a) Price comparison



(b) Price difference

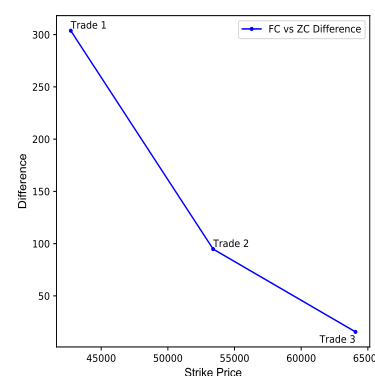


Figure 9. Zero collateral vs. fully collateralized trades.

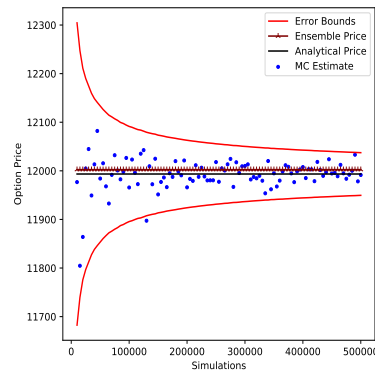
It is evident from Figure 9 that the prices generated by the bagging ensemble for a fully collateralized trade were more expensive than the corresponding zero collateral trade; thus, the first critical condition holds. The comparison of the numerical accuracy of the bagging ensemble with the Monte Carlo simulation in relation to the closed-form solution acting as the baseline price can be seen in Figure 10.

From Figure 10, it can be seen that the prices for zero collateral and fully collateralized European call options predicted by the bagging ensemble across varying levels of moneyness were within the three standard deviation error bounds of the Monte Carlo simulation up to 500,000 simulations; thus, the requirement of the second critical condition holds. As a supplementary analysis, the computation times of a single zero collateral and fully collateralized trade for the Monte Carlo simulation consisting of 500,000 simulations and the bagging ensemble are reported in Table 10.

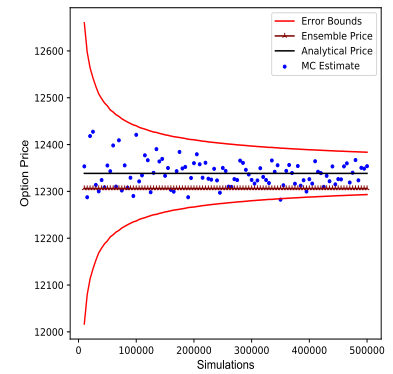
Table 10. Single trade computation time in seconds.

| Option | Monte Carlo | Bagging Ensemble |
|-------------------------------|-------------|------------------|
| Zero Collateral: Trade 2 | 0.035036 | 2.421635 |
| Fully Collateralized: Trade 2 | 0.060940 | 1.670318 |

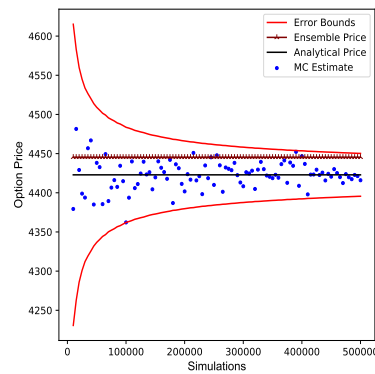
(a) Trade 1: zero collateral



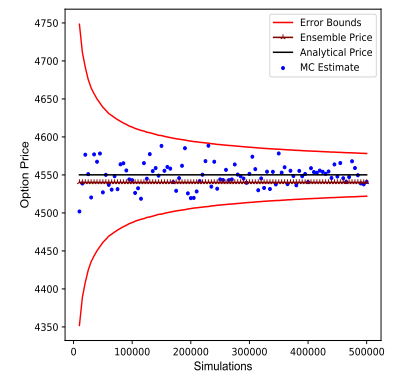
(b) Trade 1: fully collateralized



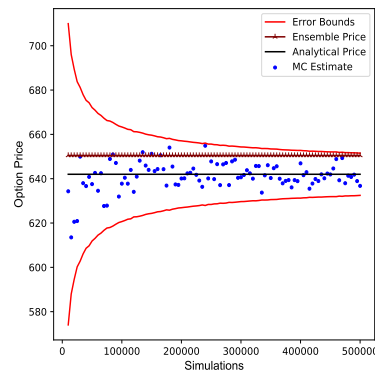
(c) Trade 2: zero collateral



(d) Trade 2: fully collateralized



(e) Trade 3: zero collateral



(f) Trade 3: fully collateralized

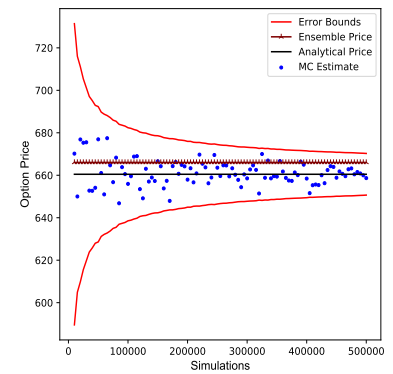


Figure 10. Comparison of trades.

From the computation times reported in Table 10, the bagging ensemble was computationally less efficient than the Monte Carlo simulation when pricing a single trade. This, however, does not necessarily translate into the bagging ensemble being computationally less efficient than Monte Carlo simulation when more than one trade is considered. To illustrate this point, the constructed implied volatility surface consisting of 10,000 implied volatility estimates in Figure 2 was converted into zero collateral and fully collateralized European call option price surfaces using the two methods. The computation times are reported in Table 11.

Table 11. Price surface computation time in seconds.

| Option | Monte Carlo | Bagging Ensemble |
|----------------------|-------------|------------------|
| Zero Collateral | 353.856228 | 18.082844 |
| Fully Collateralized | 356.446953 | 17.612219 |

It is clear from the computation times in Table 11, that the bagging ensemble was substantially faster than the Monte Carlo simulation since the Monte Carlo simulation had to generate future paths for every point on the price surfaces as each trade was treated as a separate process. The findings of this paper are concluded in the next section.

5. Conclusions

The purpose of this paper was to investigate the possibility of using machine learning techniques such as an ANN and a bagging ensemble for pricing European call options in a modern option pricing framework. By making use of an approach similar to that of Hutchinson et al. (1994), Garcia and Gençay (2000), Bennell and Sutcliffe (2004), Culkin and Das (2017), and Liu et al. (2019), it was shown that it was in fact possible to develop and train two learning networks that were able to price zero collateral and fully collateralized European call options accurately.

The bagging ensemble proved to be more accurate than the ANN in pricing zero collateral and fully collateralized European call options and was able to produce price estimates within the error bounds of the Monte Carlo simulation for trades with varying levels of moneyness in the numerical experiment. It was also shown that the bagging ensemble was computationally more efficient than Monte Carlo simulation when a large number of different trades were priced. As illustrated in Figures 6 and 8, the pricing error of the bagging ensemble can be sporadic across different levels of moneyness. Thus, for any two trades with similar characteristics, the pricing error of a given trade will differ. Nonetheless, the mere fact that it is possible to train a learning network such as a bagging ensemble with relatively few computing resources and be able to price options within the error bounds of a more well-known method such as a Monte Carlo simulation is astounding.

The results presented in this paper therefore highlighted the potential of machine learning techniques in modern financial derivative pricing problems and that the financial applications of these techniques should be further explored.

Areas for further research include investigating the performance of machine learning techniques in hedging strategies, as well as extending their applications to the pricing of exotic options.

Author Contributions: Conceptualization, R.d.P.; methodology, R.d.P. and P.J.V.; software, R.d.P.; validation, P.J.V.; formal analysis, R.d.P.; investigation, R.d.P.; data curation, R.d.P.; writing—original draft preparation, R.d.P.; writing—review and editing, P.J.V.; visualization, R.d.P. and P.J.V.; supervision, P.J.V.; project administration, P.J.V. Both authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank the editor and anonymous referees for their insightful comments and suggestions that helped improve the quality of the manuscript considerably.

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

References

- Bennell, Julia, and Charles Sutcliffe. 2004. Black–Scholes Versus Artificial Neural Networks in Pricing FTSE 100 Options. *Intelligent Systems in Accounting, Finance and Management* 12: 243–60. [CrossRef]
- Bishop, Christopher M. 2006. *Pattern Recognition and Machine Learning*. Berlin/Heidelberg : Springer.
- Black, Fischer, and Myron Scholes. 1973. The Pricing of Options and Corporate Liabilities. *Journal of Political Economy* 81: 637–54. [CrossRef]
- Breiman, Leo. 1996. Bagging Predictors. *Machine Learning* 24: 123–40. [CrossRef]
- Chollet, François . 2015. Keras. Available online: <https://github.com/fchollet/keras> (accessed on 21 November 2020).
- Culkin, Robert, and Sanjiv R. Das. 2017. Machine Learning in Finance: The Case of Deep Learning for Option Pricing. *Journal of Investment Management* 15: 92–100.
- Cybenko, George. 1989. Approximation by Superpositions of a Sigmoidal Function. *Mathematics of Control, Signals and Systems* 2: 303–14. [CrossRef]
- De Prado, Marcos Lopez. 2018. *Advances in Financial Machine Learning*. Hoboken: John Wiley & Sons.
- Garcia, René, and Ramazan Gençay. 2000. Pricing and Hedging Derivative Securities with Neural Networks and a Homogeneity Hint. *Journal of Econometrics* 94: 93–115. [CrossRef]
- Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. Cambridge: MIT Press.
- Hamori, Shigeyuki, Minami Kawai, Takahiro Kume, Yuji Murakami, and Chikara Watanabe. 2018. Ensemble Learning or Deep Learning? Application to Default Risk Analysis. *Journal of Risk and Financial Management* 11: 1–14. [CrossRef]
- Hornik, Kurt, Maxwell Stinchcombe, and Halbert White. 1989. Multilayer FeedForward Networks are Universal Approximators. *Neural Networks* 2: 359–66. [CrossRef]
- Hunzinger, Chadd B., and Coenraad C. A. Labuschagne. 2015. Pricing a Collateralized Derivative Trade with a Funding Value Adjustment. *Journal of Risk and Financial Management* 8: 17–42. [CrossRef]
- Hutchinson, James M., Andrew W. Lo, and Tomaso Poggio. 1994. A Nonparametric Approach to Pricing and Hedging Derivative Securities via Learning Networks. *The Journal of Finance* 49: 851–89. [CrossRef]
- Kingma, Diederik P., and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimisation. Available online: <https://arxiv.org/abs/1412.6980> (accessed on 4 January 2021).
- Levendis, Alexis, and Pierre Venter. 2019. Implementation of Local Volatility in Piterbarg’s Framework. In *International Conference on Applied Economics (ICOAE 2019): Advances in Cross-Section Data Methods in Applied Economic Research*. Cham: Springer, pp. 507–21.
- Liu, Shuaiqiang, Cornelis W. Oosterlee, and Sander M. Bohte. 2019. Pricing Options and Computing Implied Volatilities using Neural Networks. *Risks* 7: 1–22. [CrossRef]
- Merton, Robert C. 1973. Theory of Rational Option Pricing. *The Bell Journal of Economics and Management Science* 4: 141–83. [CrossRef]
- Piterbarg, Vladimir. 2010. Funding Beyond Discounting: Collateral Agreements and Derivatives Pricing. *Risk Magazine* 23: 97–102.
- Ruf, Johannes, and Weiguan Wang. 2020. Neural Networks for Option Pricing and Hedging: A Literature Review. Available online: <https://arxiv.org/abs/1911.05620> (accessed on 13 May 2021).
- von Boetticher, Sven T. 2017. The Piterbarg Framework for Option Pricing. Ph.D. thesis, University of Johannesburg, Johannesburg, South Africa.