International Journal of Electronic Engineering and Computer Science
Vol. 5, No. 3, 2020, pp. 22-32
http://www.aiscience.org/journal/ijeecs

Stock Prediction Using Deep Learning with Long-Short-Term-Memory Networks

Carol Anne Hargreaves*, Chen Leran

Department of Statistics and Applied Probability, National University of Singapore, Singapore

Abstract

Background: Stock market prediction is one of the most challenging tasks since the financial time series is highly volatile, noisy, non-linear and dynamic in nature. Long-Short-Term-Memory (LSTM) is a deep learning technique that focuses on sequential learning. Researchers have shown that the LSTM makes good prediction in the S&P 500 stocks, the Hong Kong stocks and the US intra-day stock data. However, there is limited research in the Australian stock market. *Objective*: This study aims to apply the LSTM technique to predict the stock price movement in the Australian Stock Market and to identify which stocks to buy for a profitable portfolio. *Methodology*: We analyzed 400 stocks and selected the top 5 stocks to buy and trade, based on the predictions of the LSTM, Regression Tree (CART) and the Auto Regressive Integrated Moving Average (ARIMA) techniques. *Results*: The results showed that the LSTM, a deep learning neural network algorithm, outperformed the CART and ARIMA-Time Series algorithms by achieving a return rate of thirty-five percent, a Sharpe ratio of 2.13 and a maximum drawdown of 0.34. The LSTM portfolio had a higher overall return rate of 35% versus the Australian market index of 21%. *Conclusion*: The LSTM networks made more accurate predictions on stock prices than the ARIMA time series and regression trees (CART). This may be because the LSTM networks is good at processing sequential data, extracting useful information and dropping unnecessary information. Further, the LSTM had a relatively more stable return compared to the ARIMA model. This is because a deep learning model is more capable of extracting the non-linear relationship in the data. In addition, the LSTM model stock portfolio outperformed the stock market index and generated profits over three consecutive time periods.

Keywords

Deep Learning, Deep Neural Network, Long-Short-Term-Memory (LSTM), ARIMA-Time Series, Classification and Regression Tree (CART), Stock Portfolio, Stock Prediction

Received: September 16, 2020 / Accepted: October 23, 2020 / Published online: December 11, 2020

1. Introduction

Since financial time series are highly volatile, noisy [11], non-linear and dynamic in nature [21], stock price prediction is one of the most challenging tasks in research [5]. To construct trading strategies based on financial modelling, it is very important for the model to be able to learn the pattern in the time series data and provide accurate predictions [20]. Over the years, many different research studies applied different techniques on the analysis of stock predictions. The techniques are mainly categorized into machine learning techniques and statistical methods [2]. In recent years, it has been proven that machine learning techniques are capable of identifying nonlinear and volatile patterns in financial time series [33].

Long-short-term-memory (LSTM) is an emerging deep learning technique. Focusing on sequential learning, it has been applied into many different research domains such as speech recognition and human action detection [14]. Meanwhile, some researchers have shown that the LSTM is capable of making good predictions in S&P 500 stocks [11], Hong Kong stocks [10] and also US intra-day stock data [6].

* Corresponding author
E-mail address: carol.hargreaves@nus.edu.sg (C. A. Hargreaves)

However, there are limited researchers applying the LSTM technique in the Australia stock market. Hence, this paper focused on the application of the LSTM technique for the prediction of stock prices in the Australia market from 2016 to 2019.

To determine whether our LSTM model performed well, we decided to use two techniques for our baseline (comparative) models, one machine learning technique and one statistical methods technique. For the machine learning technique, we considered the Classification and Regression Tree (CART) model, as it was supported by many researchers to be accurate in stock price prediction [8]. By simply using the closing prices of the stocks, the study [8] discovered the stocks with good investment value when the CART technique was applied. Further, the CART technique was modelled after the structure of a tree, and was able to provide a good explanation applicable to the prediction of the stock prices, and was able to interpret problems very much according to the principles of mathematical and statistical principles [7, 37]. The CART technique was found to be more accurate than traditional statistical methods in finding the correct relationship between the target and the feature variables [22]. Hence, the CART technique, a machine learning technique that did not focus on stock price sequence analysis, was selected as one of our baseline models. Moreover, we compared our LSTM model with a model that applied a statistical method, and chose the Auto Regressive Integrated Moving Average (ARIMA) model as our second baseline model. The ARIMA model is an algorithm commonly used in stock price prediction studies. It is widely regarded as one of the most effective models in time series forecasting [2]. Especially in short-run forecasts, the ARIMA model has a stronger predictive power than many complex structural models [1]. We hence, selected the ARIMA technique as the second baseline model for our comparison of the performance of our LSTM model.

The overall purpose of this study was to construct a Long-Short-Term-Memory (LSTM) model [6, 9, 10, 11, 18, 25] to predict the stock price movement and then compare results with two established baseline models [1, 2, 23, 28, 36]. An overview of the procedures of our study is as follows: Firstly, we use a dataset of 400 stocks. We split our data into a training dataset (2016-2018) and a testing dataset (2019). Next, each model was trained and tuned based on the training dataset. The minimization of the Mean Squared Error (MSE) was the method used for tuning the parameters. Based on the model predictions, 'k' (k is the optimised number of stocks) stocks were selected each day for the three model stock portfolios. The performance of the stock portfolios was evaluated based on the return rate (RR), maximum drawdown (MDD) and the Sharpe Ratio (SR).

The rest of this paper was organized into six sections. Section 2 presents a literature review of the LSTM, the CART and the ARIMA models. Section 3 is a description of the data and the pre-processing methods applied in this study. Section 4 presents a brief description of the application of the proposed techniques, the LSTM, the CART and the ARIMA models. Section 5 presents a description on how the portfolios are constructed and the performance of the three portfolios. The discussion for future research is covered in Section 6.

2. Literature Review

Traditional mathematical techniques have been applied to make predictions in the financial market since 1998 [31]. Stock market prediction is usually considered as one of the most challenging issues among time series predictions [36] due to its noise and volatile features. Time series methods such as the ARIMA model have been applied to financial market data to make stock predictions [28]. The Auto-Regressive Integrated Moving Average (ARIMA) is a class of models that explains a time series based on its past values, including its lags and previous error terms for forecasting. The use of the ARIMA model is under a certain degree of uncertainty because it does not make any assumptions unlike other models [34]. Although, [1] demonstrated that the ARIMA model has a robust predictive power in short-run forecasting and [2] showed that the ARIMA model was one of the most widely used statistical methods for stock price prediction. We have selected the ARIMA model for one of our baseline models.

During the past decades, machine learning models, such as Artificial Neural Networks (ANNs) [15] and the Support Vector Regression (SVR) [30], have been widely used to predict financial time series and gain high predictive accuracy [12, 19, 24, 29]. [13] showed that the random forest technique performs poorly if there are irrelevant variables in the data. Research in CART models has seen rapid growth, and applications are increasing at an even greater rate.

The Classification and Regression Tree (CART) is a decision support tool that uses a tree-like model of decisions and displays an algorithm that contains conditional control statements. CART is an umbrella term used to refer to two main types of decision trees, which are the classification tree and the regression tree. These two types of decision trees have very similar algorithms [35]. Binary decision trees are very easily interpreted in a statistical way [34]. Other than interpretability [16, 17, 27] believed that decision trees are capable of making accurate predictions in the stock price market. In a recent research [8], decision trees were demonstrated to have a strong predictive power in the US stock market, which had an accuracy rate similar to the Deep

Learning based method. Interpretability of the tree structures is a strong reason for their popularity among practitioners [23].

In recent years, after the LSTM Recurrent Neural Network had been proposed, it gained popularity with financial researchers [18], as the LSTM model can process not only single data points (images), but also the entire sequences of data (videos, time series, etc). Just like normal recurrent neural network frameworks, the LSTM model is able to learn data through one or multiple layers [25]. LSTM networks are suitable for making predictions based on time series data, since there can be lags of unknown duration between important events in a time series and were developed to deal with the exploding and vanishing gradient problems that can be encountered when training traditional Recurrent Neural Networks (RNNs). Learning in LSTM is very time consuming compared to machine learning methods [25].

Several researchers used LSTM models for predicting stock prices, including [11], who applied LSTM network to daily prices of the S&P 500 stocks from 1992 to 2015. Their approach included the standardization of the daily return of all stocks and trained the LSTM model with three years of data and tested it in the next year. In their studies [11], there is an obvious outperformance of the LSTM model over the random forest, the deep neural network model and the logistic regression classifier. Other than that, a study using the LSTM-based method conducted by [9] demonstrated that the daily stock prices in the Chinese Stock Market had accurate predictions. However, to our knowledge, the application of the LSTM model in the Australian stock market is not well developed yet, and hence, we focused on the application of the LSTM model to predict stock price movement as the main focus of this paper.

On the whole, machine learning and deep learning techniques were proven useful in stock price predictions by connecting the nonlinear relationships between the stock price and the predictors. These techniques include the support vector machine method (SVM) [28], the random forest method [4], the neural network technique [32] and the LSTM [11]. This study builds a novel forecasting framework to predict the one-step-ahead closing price of six popular stock indices traded in different financial markets. Further, this study focuses on the stock price prediction using the LSTM technique and compares the LSTM model performance with the CART and the ARIMA model performances.

3. Data Description and Data Pre-processing

Stock price data from Australian Stock Market (2016 - 2019) was used for this study. The stocks were from different industries including Utilities, Industrial, Finance, Energy, Information Technology, etc. However, some stocks did not have complete data from 2016 to 2019. This is because these companies did not exist for the entire period understudy. Data processing and modelling was performed using Python (3.7). The packages used included numpy, pandas, tensorflow, sci-kit and pyramid-arima. The LSTM network was trained on NVIDIA-GPUs. All data before 2019 was used for training the model, and all stock data in 2019 was used for testing the model performance.

When there were missing values in the stock price data, the null value was replaced with the previous day's stock price. As part of the data cleaning process, stocks with more than 200 stock price's the same as the previous day's stock price, were excluded from the modelling dataset. The final modelling dataset had 400 stocks, represented in 300 000 rows of data.

For each stock i, on day t, the adjusted close price was denoted by P_i^t and the daily log return on day t for stock i was calculated by

$$\log (R^i_t) = \log_{P^i_{t-1}} P^i_t \tag{1}$$

As shown in equation 1, the return for stock i on day t is denoted by R^i_t. Hence, to predict the return of stock i on day t, if the return of the past l days is considered, the feature vector is R^i_{t-1}, R^i_{t-2}, ……, R^i_{t-l} where l is named as the 'lookback', and refers to the number of past days' return. The 'lookback' period, l, is one of the parameters of the LSTM model that is tuned to obtain an optimized stock price prediction.

Each stock has a feature vector is R^i_{t-1}, R^i_{t-2}, ……, R^i_{t-l} every day. By stacking all these vectors together, a large feature vector V is constructed. Vector V has dimensions equal to (number of stocks * length of the training window). The stock returns from the training dataset are standardized by subtracting the average (μ_{train}) and dividing by the standard deviation (σ_{train}).

$$R^{i*}_t = (R^i_t - \mu_{train}) / (\sigma_{train} \tag{2}$$

The response variable for each stock i and day t is simply the value of R^{i*}_t.

4. Methodology

This section briefly introduces the three models understudy for this paper, the LSTM model, the CART model and the ARIMA model.

4.1. Long-Short-Term-Memory (LSTM)

The Long-short-term-memory (LSTM) model has an artificial recurrent neural network (RNN) architecture. Traditional neural networks take in the features as single data points, however, the LSTM network take in the features as sequences of data, such as speech, video and time series data. Similar to traditional RNNs, each LSTM network composes of an input layer, a hidden layer(s) and an output layer. Unlike RNNs, the hidden layer(s) of the LSTM are constructed as memory cells. Each memory cell consists of a forget gate, an input gate and an output gate. The three gates work together to regulate the flow of information and to discard unimportant information between memory cells.

1) Forget gate (f_t): Defines which information to be removed from the cell state

2) Input gate (i_t): Defines which information to be added to the cell state

3) Output gate (o_t): Defines which information to be used as an output

Figure 1 below, illustrates the flow of information at time t. Each gate was considered as a neuron in a multi-layer neural network. Each gate was associated with an activation function to compute a weighted sum. The 3 exit arrows from the memory cell to the 3 gates represent the peephole connections. They denote the contributions of the activation of the memory cell c, at time t-1. It means that the calculations made at step t also consider the activation of the memory cell at time step t-1.

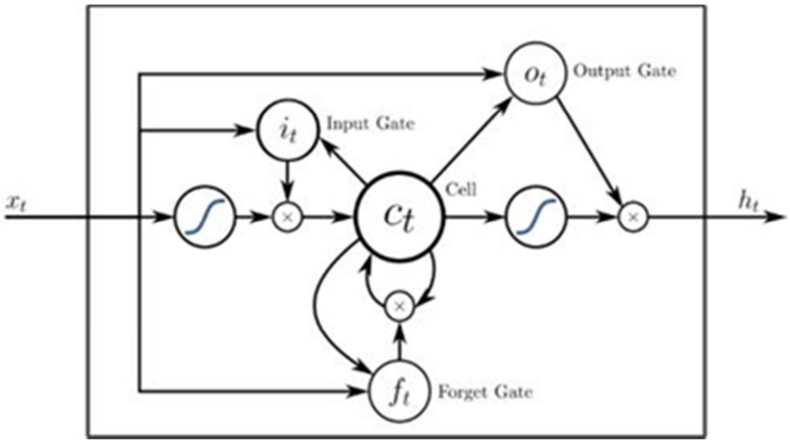

Figure 1. The Architecture of the LSTM Network.

The equations and calculations are listed below:

$$f_t = \sigma_g (W_f x_t + U_f h_{t-1} b_f) \quad (3)$$

$$i_t = \sigma_g (W_i x_t + U_i h_{t-1} + b_i) \quad (4)$$

$$o_t = \sigma_g (W_o x_t + U_o h_{t-1} + b_o) \quad (5)$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \sigma_c (W_c x_t + U_c h_{t-1} + b_c) \quad (6)$$

$$h_t = o_t \circ \sigma_c(c_t) \quad (7)$$

Where the meaning of the notations are:

1) $x_t \in R^d$: input vector to the LSTM unit at time t.

2) $f_t \in R^h$: forget gate's activation vector

3) $i_t \in R^h$: input gate's activation vector

4) $o_t \in R^h$: output gate's activation vector

5) $h_t \in R^h$: hidden state vector also known as output vector of the LSTM unit

6) $c_t \in R^h$: cell state vector

7) $W_t \in R^{hxd}$, $U_t \in R^{hxh}$: weight matrices learnt in training

8) b_f; b_i; $b_o \in R^h$: bias vector parameters learnt in training

The activation functions involved in the equations and the definitions are explained as follows. The completely forget activation function scales the activation values to be within the range between 0 and 1 and the completely remember activation function scales the activation values to be within the range between -1 and 1.

1) σ_g: sigmoid function = $e^x / (e^x + 1)$

2) σ_c: hyperbolic tangent function $\sigma_c(x) = (e^{2x} - 1) / (e^{2x} + 1)$

3) o: Hadamard (elementwise) product

At time t, firstly, the LSTM layer takes in the information from the previous cell state c_{t-1}. The activation values f_t of the forget gates at time t are computed based on x_t, h_{t-1} and b_f. Secondly, the LSTM network determines which information should be added to the current cell state c_t by computing i_t. In addition, $\sigma_c (W_c x_t + U_c h_{t-1})$ in Equation 6 refers to the value which can be potentially

added to the cell states. Thirdly, c_t was calculated as the sum of elementwise product of each part. Finally, the output h_t was computed. Just like other machine learning methods, the weight matrices W_s and the bias vectors b_s were trained in the iterations such that they minimized the loss function, Mean

Squared Error (MSE) across the training samples.

The parameters we considered for tuning were the epochs, lookback, memory units and batch size. The initial model parameters used was, number of Epochs =150, Lookback period = 252 days, memory unit = 20 and batch size =1. The parameter definitions and tuning procedures are listed below.

1) Epochs: The number times that the learning algorithm will work through the entire training data set. If the number of Epochs were not enough, there were potential improvements for the model to reach higher accuracy. With increased Epochs, the training Minimum Squared Error (MSE) decreased because the model fits the training data closer. However, if the number of Epochs was too large, this indicated that the training model was overfitted. Initially, the number of Epochs was set at 150, which is quite a large number, so that the change in the MSE over the Epochs was observed. Once the MSE stopped decreasing significantly, a suitable number of Epochs was determined. The number of Epochs was determined as 140.
2) Lookback: The length of timestamps passed to the model. If the Lookback period is too short, not enough information is given to the model. If the Lookback period is too large, irrelevant data will be passed to the model and this will affect the model performance. The initial model set up was with a Lookback = 252, which is roughly the number of trading days in a year. After fine tuning the Lookback parameter, the Lookback was set at 30.
3) Memory Units: LSTMs are recurrent networks where each neuron is replaced by a Memory Unit. The Memory Unit contains an actual neuron with a recurrent self- connection. The initial number of Memory Units was set at 20. The lowest MSE occurred at Memory Units equal to 1.
4) Batch Size: The Batch Size defines the number of samples to use for fine tuning the parameters. On sequence prediction problems, it may be desirable to use a large Batch Size when training the network and a Batch Size of 1 when making predictions for the next step in the sequence, so the initial Batch Size value was set as 1. The lowest MSE occurred at Batch Size equal to 1.

A grid search to find the optimal combination of the of the Memory Unit and the Batch Size was performed. A heat map of the results is shown in table 1 below.

Table 1. The Heat Map of the Grid Search for Optimal Parameters.

batchsize \ memory unit	1	2	3	10
1	0.0378	0.0383	0.0382	0.0404
2	0.0380	0.0385	0.0391	0.0413
3	0.0381	0.0386	0.0385	0.0409
4	0.0383	0.0387	0.0389	0.0412

The grid search result was in agreement that the most suitable setting for both the Memory Unit and Batch Size was 1 as the MSE had a global minimum of 0.0378 at Memory Unit =1 and Batch Size =1. Hence, after tuning all the parameters, the model set up was Epochs = 140, Lookback = 30, Memory Units = 1 and Batch Size = 1.

4.2. The Classification and Regression Tree (CART)

The Classification and Regression Tree (CART) is derived from splitting the training observations into a tree-like fashion. The mean of the training observations *i*n each Lookback period is used as the prediction for that time period. The regression tree was built as follows:

1. The predictor space (i.e. the possible values of (x1, x2, x3, …xN) was divided into J distinct non-overlapping regions $R_1, R_2,..., R_J$.
2. For every observation that falls in the region Rj, the same prediction is made, which is simply the mean of the response values for the training observations in R_j.

Under this setup, the model aims to find boxes $R_1, R_2,..., R_J$ such that

$$\sum_{J-1}^{J} \sum_{i \in R_j} (y^i - y^*_{Rj})^2 \quad (8)$$

was minimized, where $y_{Rj}*$ is the predicted value for the region R_j. The splitting was done in a recursively binary manner. The trees were split in a top down and greedy approach. Top down because the splitting begins at the top of the tree and successively splits the predictor space. The approach is greedy because it takes the best split at each step, with no consideration for the future.

In the first step, we considered all predictors $X_1,..., X_p$ and all possible values of the cut-point s for each of the predictors. That is, we have candidates

$$R_1(j, s) = \{X \mid X_j < s\} \text{ and } R_2(j, s) = \{X \mid X_j\} >= s\} \quad (9)$$

and choose the pair (j, s) that minimizes Equation 8. Then this binary split is repeated recursively to find the next best split. The model for each stock was trained separately because the features included in the CART model involved the standard

The features included in the Regression Tree were as follows:

1) Volume 1: Yesterday's volume of the stock. It indicates the availability of the stock in the market and how active it is.
2) Standard Deviation: It indicates the volatility of the stock price.
3) Exponential Moving Average: It reflects the weighted average of the price level over the past days.

For each stock the Regression Tree for was trained and pruned by cross validation. To calculate the overall variable of importance, the average of variable contribution across all stocks was used. For example, the importance of Volume 1 is the aver-age importance contributed for each of the models. Volume 1 was the most important feature, followed by the Standard Deviation over 5 days, 20 days, 10 days 15 days and then the Exponential Moving Average over 15 days. The Regression Tree model selected the volatility indicator as the more important feature than the price level indicator.

4.3. The Auto-regressive Integrated Moving Average (ARIMA)

An Autoregressive Integrated Moving Average (ARIMA) Model is a statistical model for analyzing and predicting time series data. Simply as the name suggests, it consists of three key aspects:

AR: Autoregression. A model that uses the dependent relationship between an observation and some number of lagged observations. It is associated with a parameter p, which is the number of lag observations included in the model. The AR(p) model is defined as

$$X_t = c + \sum \phi_i X_{t-i} + \epsilon_t \text{ for } i = 1, 2, 3, \ldots p \quad (10)$$

where ϕ_i's are the parameters, c is a constant, t represents stock t and ϵ_t is a white noise.

I: Integrated. The use of differencing of raw observations (e.g. subtracting an observation from an observation at the previous time step) in order to make the time series stationary. The parameter, d indicates the number of times that the raw observations are differenced, also called the degree of differencing. For example, an ARIMA(1,1,0) model is expressed as

$$X_t - X_{t-1} = c + \phi (X_{t-1} - X_{t-2}) \quad (11)$$

MA: Moving Average. A model that uses the dependency between an observation and a residual error from a moving average model applied to lagged observations. It is associated with a parameter q, which is the size of the moving average window. The MA(q) model is defined as

$$X_t = \mu + \sum \theta_i \epsilon_{t-i} + \epsilon_t \text{ for } i = 1,2,3 \ldots\ldots, q. \quad (12)$$

where μ is the mean of the time series, θ_i's are the parameters of the model and the ϵ_t's are the white noise error terms.

A linear model is constructed based on the orders of p, d and q. An ARIMA (p, d, q) model is given by

$$\Phi (B)(1 - B)^d X_t = \theta (B) \epsilon_t \quad (13)$$

where

$$B X_t = X_{t-1} \quad (14)$$

$$\varphi(B) = 1 - \varphi_1 B - \varphi_2 B - \cdots - \varphi_p Bp \quad (15)$$

$$\theta(B) = 1 - \theta_1 B - \theta_2 B_2 - \cdots - \theta_q B^q \quad (16)$$

Unlike the set up in LSTM networks, where the vectors of all stocks are stacked together to construct the feature space, the ARIMA model is specifically for time series only. Hence, each stock had its own ARIMA model. Moreover, the stock prices were the inputs for the ARIMA model. As the stock prices were daily data without seasonality, the criterion of Akaike information criterion (AIC) was applied for the order selection. It is defined as

$$AIC = - 2 \log(L) + 2(p + q + k) \ldots. \quad (17)$$

where L is the likelihood of the data, p is the order of the AR part and q is the order of the MA part. The k represents the intercept of the ARIMA model. The best models were selected by minimizing the AIC. Moreover, when fitting the time series data to the ARIMA model, the parameters were tuned in a stepwise manner to reduce the possibility of overfitting the model.

The ARIMA model was built on a rolling basis every week for each stock, as shown in figure 2 below:

Figure 2. ARIMA Model Rolling Training Window.

5. Results and Discussion

This section describes the portfolios constructed by the results of the models. There were 400 stocks to select from and form a portfolio each trading day. In this study, the portfolios were constructed by ranking the prediction for all stocks each day and then selecting the stocks with the highest prediction. Which means that, from LSTM networks and CART models, the stocks are simply ranked by the predicted values because the targets of these two models are return rates together with their respective probabilities. For the ARIMA model, the predicted return rate is calculated first based on the model output because the target in this model is the price itself. Moreover, if 5 stocks are used for the portfolio, then we open a long position for the 5 stocks on that specific day. On the next day, if 3 out of 5 stocks have positive returns or have the same price as the previous day, we hold these three stocks, and open a long position for two new stocks and sell the two stocks that had a negative return. See figure 3 below for an illustration of our portfolio construction steps.

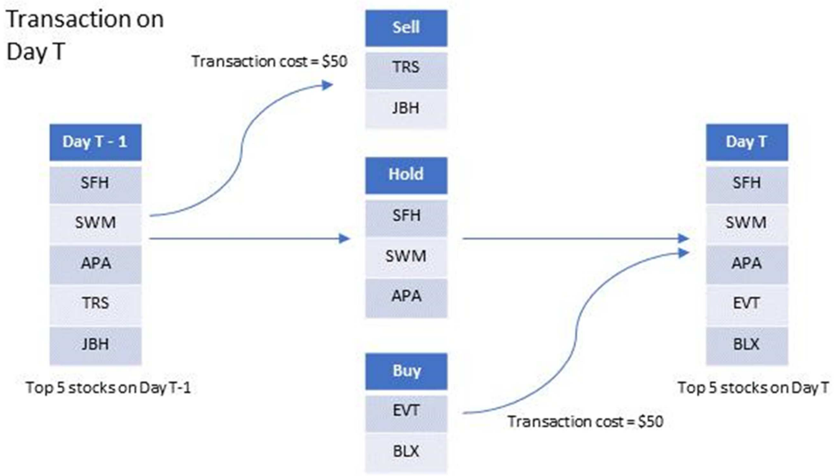

Figure 3. Portfolio Construction Steps.

The price range of the stocks was from 0.01 to 200 dollars. We chose to invest in a notional equal manner. We fixed the total investment on each stock at $20,000. Other than the limitation of investment on each stock, in the case where the price stock was too small, such as $0.01, it is impossible to buy 2,000,000 shares of a certain stock because it will not be available in the market. In this case, we added a threshold to the number of shares invested as well. The total number of shares invested on each stock should not exceed 25,000. Other than the cost of stocks, we also needed to consider the transaction costs. In this study, the transaction cost was set at $25. Whenever a stock was purchased or sold, a cost of $25 was incurred. For example, on the first day, if 10 different stocks were purchased, 250 dollars was paid as a commission fee. If 10 stocks were held on day t-1 and on day t, 2 of the top 10 stocks were different to yesterday, then a commission of 2 * 25 * 2 = $100 was paid for both entering and exiting the market.

Next, we needed to determine how many stocks to trade each day. If too few stocks were selected, the stock portfolio would lack diversification. If too many stocks were selected, then the advantage of selecting stocks with the highest prediction would fade away. Another issue of selecting too many stocks to trade each day is that the total investment and transaction costs would be too high. Hence, the need to determine an optimum k, the number of stocks to be selected for each stock portfolio.

To select the optimum k, the k with the highest portfolio return per dollar of investment. was chosen. Figure 4 shows the return rate, total profit before transaction costs and total profit after transaction costs according to the result of each model, and also the comparison of the return rate of each model. It is shown clearly that the performance of regression tree is not as good as the other two models. Focusing on the rate of return only, the portfolio selected by ARIMA has the best rate of return at k = 1; 2; 3 and the portfolio selected by LSTM has the best rate of return at k = 4; 5. Let's focus on these portfolios with the best rate of return.

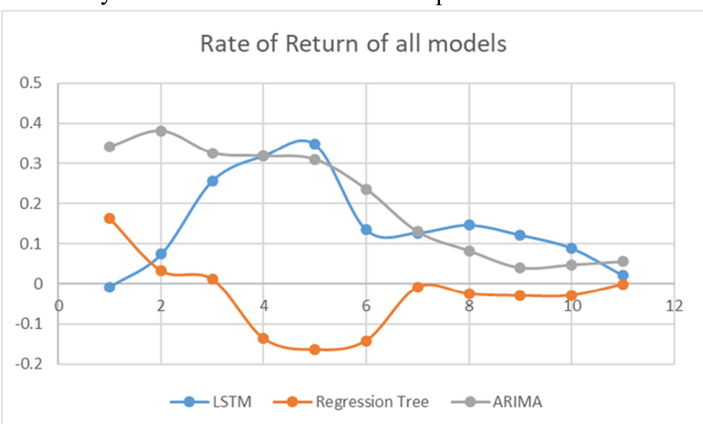

Figure 4. Finding the Optimum k based on Rate of Return of all models.

Table 2 includes the statistics of each portfolio. Rate of return is the annualized return rate. Average Daily PnL is the Profit and Loss, calculated in dollars. Std is the standard deviation of each portfolio in one year. It indicates the volatility of the portfolio. Maximum drawdown (MDD) is the maximum observed loss from a peak to a trough. It is an indicator of downside risk. Usually a portfolio with a higher MDD is riskier. Sharpe ratio is the average return earned in excess of the risk-free rate per unit of volatility or total risk.

Table 2. Key Descriptive Statistics of Each Portfolio.

Portfolio	Rate of Return	Investment	Average Daily PnL	Std	MDD	Sharpe Ratio
LSTM k=4	0.32	108,353	140.60	0.0131	0.30	1.29
LSTM k=5	0.35	114,493	161.86	0.0136	0.34	2.13
ARIMA k=1	0.34	23,709	32.95	0.0425	1.85	0.88
ARIMA k=2	0.38	59,837	92.69	0.0228	0.85	1.01
ARIMA k=3	0.33	75,555	100.07	0.0207	1.87	0.76

In Table 2, all portfolios have annualized returns greater than 30%. However, the standard deviation is quite different. For the portfolio [ARIMA k = 1], the standard deviation is the highest. This is possibly because only one stock is selected to buy each day, and the portfolio lacks diversification. Other than this portfolio, generally all portfolios selected by the ARIMA model have a larger standard deviation, a larger MDD per dollar of return and a smaller Sharpe ratio. Despite the similar rate of return, we conclude that the portfolios based on the ARIMA models were more volatile. Moreover, as shown in Figure 5 below, from August onwards, all portfolios selected by ARIMA model has a negative return. Unlike portfolio [LSTM k=4] and [LSTM k=5] which both have an increasing trend across the year.

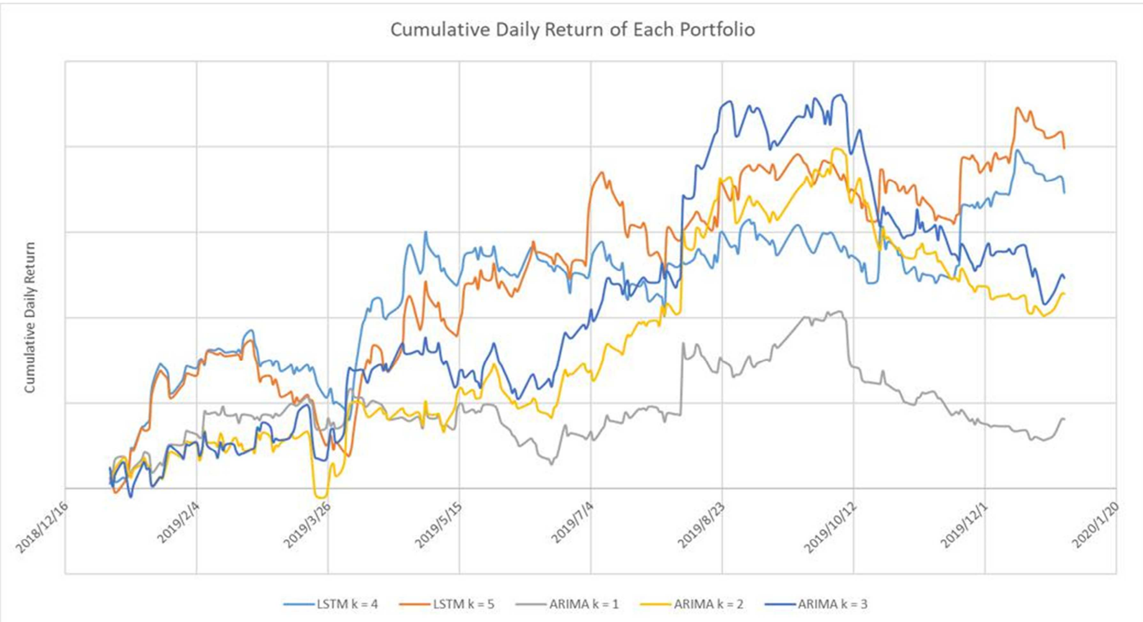

Figure 5. Cumulative Daily Returns of the Stock Portfolios.

Therefore, considering the maximum drawdown and Sharpe ratio of each portfolio, and whether the portfolio has a relatively steady performance, the portfolio given by [LSTM k = 5] was finally chosen as the best performing model as it had a Sharpe ratio of 2.13, which was the highest among all the portfolios.

We next, focused on portfolio [LSTM k = 5], and conducted a three-period analysis. The one year testing period was divided into three equal length periods to analyze the stability of the portfolio. Moreover, the portfolio performance was compared with the ‘All Ordinaries Index’ (AˆORD), the Australian stock market index to determine whether our portfolios outperformed the Australian stock market index.

As shown in Table 3 below, the investment for each quarter was capped at $115,000. Although the rate of return in the third quarter was lower than the first quarter, there was still a positive return. Moreover, the portfolio had a higher return rate than the stock market index in three consecutive periods. Even though in the third quarter, the portfolio had a return of only 2.63%, it was still higher than the stock market index return of 1.87% in the same period. The stock portfolio had a weak performance in the third quarter probably due to the weak performance of the overall stock market.

Table 3. Three Period Analysis for [LSTM k = 5] and AˆORD.

Time Period	Investment	PnL	Rate of Return	A^ORD Start of Period	A^ORD End of Period	Rate of Change
2019 Jan - April	100,590.82	18,612.23	18.50%	5625.6	6418.4	14.09%
2019 May - Aug	88,863.06	18,194.81	20.48%	6466.5	6698.2	3.58%
2019 Sept - Dec	114,492.53	3,009.37	2.63%	6677.5	6802.4	1.87%

In Figure 6 below, the total value of the portfolio and the AˆORD index are displayed. The upper limit of each axis equals to 1.5 times that of the beginning value such that the gradient for the two line charts are comparable. Figure 6 shows that generally the stock portfolio outperformed the market index. The LSTM portfolio had a higher overall return rate of 35% versus the Australian market index of 21%.

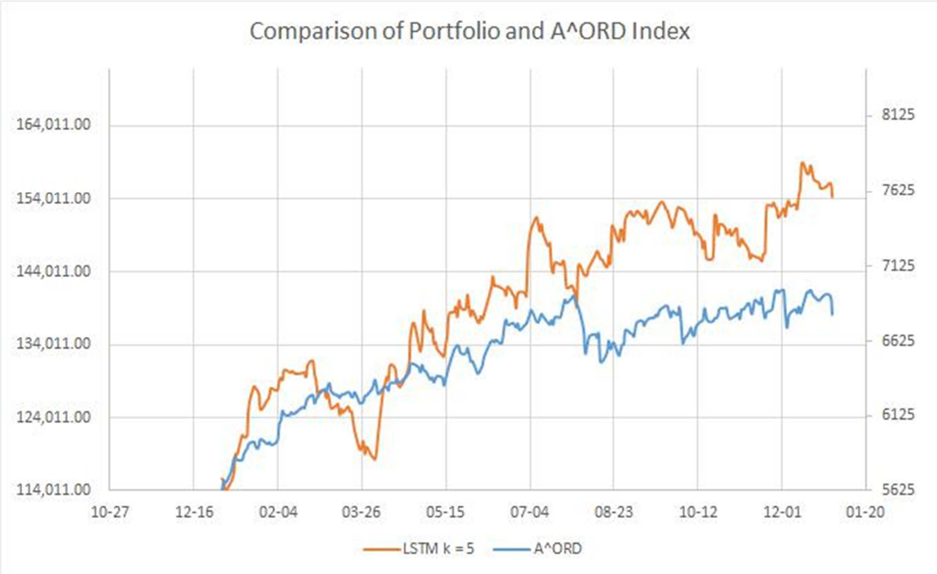

Figure 6. Comparison of the Stock Portfolio and AˆORD Index performance.

6. Conclusion

The ARIMA model is one of the many traditional methods that rely on an underlying stochastic model to extract information from the data, while the CART technique is modelled after the structure of a tree, and provides an explanation to the prediction of the stock prices. The ARIMA model outperformed the CART model.

The LSTM model represents an algorithmic approach to analysing time series data and treats the underlying process as unknown. In this study, our LSTM model was able to make more accurate predictions on stock price movements compared to the CART model. This is because the LSTM model, by its nature, uses a deep learning approach and is good at processing sequential data, and extracting useful information while dropping unnecessary information.

In addition, a three period analysis showed that the LSTM stock portfolios produced positive returns over all three consecutive time periods, demonstrating a robust, stable methodology for producing profitable returns.

In conclusion, the portfolio selected by our LSTM network, had a relatively more stable return compared to the ARIMA model. This is because a deep learning model is more capable of extracting the nonlinear relationship in the data compared to traditional linear statistical models.

From a practical point of view, for investors interested in choosing between statistical methods, or machine learning techniques, or deep learning techniques for determining which stocks to buy, we suggest investors use the LSTM model for their stock price predictions because of the deep learning approach of the LSTM model. With the deep learning approach, the LSTM model outperformed the ARIMA, and CART model.

For future research, we note that our LSTM model performed well over a short-term forecasting horizon, example, one-step-ahead and we did not consider medium and long term horizons. We therefore suggest further research be performed to determine the performance of the LSTM networks for the medium and long term horizons. Further, our results demonstrated that the LSTM model outperforms the CART and ARIMA models for the Australian stock market. It is suggested that further research be performed to better understand the performance of the LSTM model in other geographic locations.

References

[1] Adebiyi, A. A., Adewumi, A. O., Ayo, C. K (2014). Stock Price Prediction using the ARIMA Model. UKSim-AMSS 16th International Conference on Computer Modelling and Simulation, DOI 10.1109/UKSim.2014.67, pages, 105 - 111

[2] Adebiyi, A. A., Adewumi, A. O., Ayo, C. K (2014). Comparison of ARIMA and artificial neural networks models for stock price prediction. *Journal of Applied Mathematics*, Volume 2014, Article ID 614342, 7 pages, https://www.hindawi.com/journals/jam/2014/614342/

[3] Ballings, M., Van den Poel, D., Hespeels, N., Gryp, R (2015). Evaluating multiple classifiers for stock price direction prediction. *Expert Systems with Applications*, Vol. 42, page, 7046–7056.

[4] Bao, W., Yue, J. and Rao, Y (2017). A deep learning framework for financial time series using stacked autoencoders and long short term memory. PLoS ONE, Vol. 12 (7): e0180944. https://doi.org/10.1371/journal.pone.0180944

[5] Barak, S., Modarres, M (2015). Developing an approach to evaluate stocks by forecasting effective features with data mining methods. *Expert Systems with Applications*, Vol. 42, page, 1325–1339.

[6] Boroovkova, S., Tsiamas, I (2019). An ensemble of LSTM neural networks for high-frequency stock market classification. *Journal of Forecasting*, v. 38, page, 600–619. DOI: 10.1002/for.2585

[7] Brida, J. G., & Risso, W. A. (2010). Hierarchical structure of the German stock market. *Expert Systems with Applications*, Vol. 37(5), pages, 3846–3852.

[8] Chang, T. S (2011). A comparative study of artificial neural networks, and decision trees for digital game content stocks price prediction. *Expert Systems with Applications*, Vol. 38, page, 14846–14851.

[9] Chen, K., Zhou, Y., Dai, F (2015). A LSTM-based method for stock returns prediction: A case study of china stock market. *2015 IEEE International Conference on Big Data (Big Data)*, Santa Clara, CA, page. 2823–2824.

[10] Chen, S., Ge, L (2019). Exploring the attention mechanism in LSTM-based Hong Kong stock price movement prediction. *Quantitative Finance*, Vol. 19, pages, 1507–1515. DOI: 10.1080/14697688.2019.1622287.

[11] Fischer, T., Krauss, C (2018). Deep learning with long-short-term-memory networks for financial market predictions. *European Journal of Operational Research*, Vol. 270, pages, 654–669.

[12] Gao, P., Zhang, R, Yang, X (2020). The Application of Stock Index Price Prediction with Neural Network. Mathematical and Computational Applications — Open Access Journal, 25, 53; doi:10.3390/mca25030053

[13] Gashler, M., Giraud-Carrier, C. G. & Martinez, T. R. (2008). Decision tree ensemble: small heterogeneous is better than large homogeneous. In *Seventh International Conference on Machine Learning and Applications*, pages, 900–905. Washington, DC: IEEE Computer Society.

[14] Graves, A., Liwicki, M., Fernandez, S., Bertolami, R., Bunke, H., Schmidhuber, J (2009). A novel connectionist system for unconstrained handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 31, No.5, pages, 855-868.

[15] Guo Z, Wang H, Liu Q, Yang J (2014). A Feature Fusion Based Forecasting Model for Financial Time Series. *Plos One*, Vol 9(6), pages, 172-200.

[16] Hargreaves, C. A (2017). Machine Learning Application in the financial Markets Industry. *Indian Journal of Scientific Research,* Volume 17, Issue 1, pages, 253-256.

[17] Hargreaves, C. A., Yi, H (2012). Does the use of Technical & Fundamental Analysis improve Stock Choice?: A Data Mining Approach applied to the Australian Stock. Market. IEEEXPLORE Conference Proceeding. https://ieeexplore-ieee-org/document/6396537

[18] Hochreiter, S., Schmidhuber, J (2006). Long short-term memory. *The MIT Press Journals*, Vol. 9, pages, 1735–1780. https://doi-org.libproxy1.nus.edu.sg/10.1162/neco.1997.9.8.1735

[19] Huang, W., Nakamori, Y., Wang, S. Y (2005). Forecasting stock market movement direction with support vector machine. *Computers & Operations Research.* Vol. 32(10), Pages, 2513-2522.

[20] Kim, K-J, Han, I (2000). Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. *Expert Systems with Applications*, Vol. 19, pages, 125–132. https://doi.org/10.1016/S0957-4174(00)00027-0

[21] Lahmiri, S (2012). Linear and nonlinear dynamic systems in financial time series prediction. Management Science Letters 2, pages, 2551–2556

[22] Levin, N, Zahavi, J (2001). Predictive modeling using segmentation. Journal of Interactive Marketing, Vol. 15, pages. 2–22.

[23] Loh, W.-Y (2014) Fifty years of Classification and Regression Trees. *International Statistical Review*, Vol. 82, Issue 3, pages, 329–348. doi:10.1111/insr.12016

[24] Lu, C. J., Lee, T. S., Chiu, C. C (2009). Financial time series forecasting using independent component analysis and support vector regression. Decision Support Systems. Vol. 47(2), pages, 115-125.

[25] Lui, Y (2019). Novel volatility forecasting using deep learning long-short-term-memory recurrent neural networks. *Expert Systems with Applications*, Vol. 132, pages. 99–109.

[26] Mallikarjuna, M, Prabhakara Rao, R (2019). Evaluation of forecasting methods from selected stock market returns. Financial Innovation. https://jfin-swufe.springeropen.com/articles/10.1186/s40854-019-0157-x

[27] Ou, P, Wang, H (2009). Prediction of stock market index movement by ten data mining techniques. *Modern Applied Science*, Vol. 132, pages. 28–42.

[28] Pai, P.-F., Lin, C.-S (2005). A hybrid ARIMA and support vector machines model in stock price forecasting. *Omega*, Vol. 33, pages. 497–505.

[29] Prasaddas, S., Padhy, S (2012). Support Vector Machines for Prediction of Futures Prices in Indian Stock Market. *International Journal of Computer Applications.* Vol. 41(3), pages, 22-26.

[30] Pyo, S, Lee, J, Cha, M. Jang, H (2017). Predictability of machine learning techniques to forecast the trends of market index prices: Hypothesis testing for the Korean stock markets. PLoS ONE 12(11): e0188107. https://doi.org/10.1371/journal.pone.0188107

[31] Stoean, C, Paja, W, Stoean, R, Sandita, A (2019). Deep architectures for long-term stock price prediction with a heuristic-based strategy for trading simulations. https://doi.org/10.1371/journal.pone.0223593.

[32] Singh, R.; Srivastava, S (2017). Stock prediction using deep learning. *Multimedia Tools and Applications,* Vol. 76, pages. 18569–18584.

[33] Tabachnick, B. G., Fidell, L. S (2013). *Using Multivariate Statistics*. Upper Saddle River, NJ, USA, 6th edition.

[34] Takeuchi, L., Lee, Y.-Y (2013). *Applying deep learning to enhance momentum trading strategies in stocks*.

[35] Ture, M., Tokatli, F., Kurt, I (2009). Using Kaplan–Meier analysis together with decision tree methods in determining recurrence-free survival of breast cancer patients. *Expert Systems with Applications*, Vol. 36, pages, 2017–2026.

[36] Wang, B., Huang, H., Wang, X (2012). A novel text mining approach to financial time series forecasting. *Neurocomputing*, Vol. 83(6), pages, 136-145.

[37] Yin, Q, Zhang, R, Liu, Y, Shao, X (2017). Forecasting of stock price trend based on CART and similar stock. International Conference on Systems and Informatics (ICSAI). DOI: 10.1109/ICSAI.2017.8248523