Comparitive Study of Time Series and Deep Learning Algorithms for Stock Price Prediction

Santosh Ambaprasad Sivapurapu
Department of Applied Mathematics
Liverpool John Moores University
London, Unites Kingdom

***Abstract*—Stock Price Prediction has always been an intriguing research problem in financial domain. In the past decade, various methodologies based on classical time series, machine learning, deep learning and hybrid models which constitute the combinations of algorithms have been proposed with reasonable effectiveness in predicting the stock price. There is also considerable research work in comparing the performances of these models. However, from literature review, stems a concern, that is, lack of formal methodology that allows comparison of performances of the different models. For example, the lack of guidance on the generalizability of the time series models and optimised deep learning models is concerning. In addition, there is also a lack of guidance on general fitment of models, which can vary in accordance with forecasting requirement of stock price. This study is aimed at establishing a formal methodology of comparing different types of time series forecasting models based on like for like paradigm. The effectiveness of Deep Learning and Time-Series models have been evaluated by predicting the close prices of three banking stocks. The characteristics of the models in terms of generalizability are compared. The impact of the forecasting period on performance for various models are evaluated on a common metric. In most of the previous studies, the forecasting was done for the periods of 1 day, 5 days or 31 days. To keep the impact of volatility in the stock market due to various political and economic shocks both at international and domestic domains to the minimum, the forecasting periods of up 2 days for short term and 5 days for long term are considered. It has been evidenced that the deep learning models have outperformed time series models in terms of generalisability as well as short- and long-term forecasts.**

Keywords*—*Time series; deep learning; ARIMA; VAR; LSTM; GRU; CNN 1D; genetic algorithm; Tree Structured Parzen Estimator (TPE)

I. Introduction

The stock markets and associated indexes are considered as one of the important economic indicators of the state. The movement of the stock price is considered as a representation of the confidence that businesses are entitled to. Likewise, a healthy stock market movement indicates a general positive confidence in the economy. A steady and upward increase of the stock price of a business indicates the progression of business in the right direction. And such circumstances provide businesses an opportunity to use stock market as a sustainable and economical source of raising capital for further investments and growth. This cycle of business investment and successive growth assuredly brings monetary benefits to investors.

The other category of market players who materialise the stock price movements into monetary benefits are the traders. Traders exploit the crests and troughs of the stock price movement by buying and selling the shares for profit, in addition to hedging their bets. Forecasting the stock prices allow traders to make an informed decision thus, optimising the trading strategies and in turn maximising the benefits.

The stock price movement is essentially a culmination of multitudinal events including human sentiments. The randomness in the sticker movement coupled with influence of numerous exogenous variables, which often represent global macro-economic events poses unique challenges in building reasonable predictive machine. However, the monetary benefits from the returns of stock market investments and the abundance of data availability, in addition to the technical advances in hardware acceleration motivates constant research in this domain. From the perspective of implementing the predictive model, there are problems associated with selection of the algorithms for effective accuracy with the available data, how far the forecasting period can be extended while the accuracy of prediction is within operational requirements and what kind of algorithms are generalizable. This research aims to solve this problem. The overall study is structured into following broad objectives.

Study the effectiveness in forecasting: Both classical time series and deep learning models will be trained using the same stock data and the measure of accuracy in forecasting the stock prices is compared. The feature sets used for training constitute the daily prices of the stock, the technical features of the respective stock and macro-economic indicators termed as exogenous variables henceforth.

Study the generalizability of the models: Both time series and deep learning models will be evaluated on different banking stocks and the effectiveness of the prediction will be compared to understand the generalizability of the models.

Study the impact of forecasting periods on accuracy: Both the time series and deep learning models will be trained and used to forecast the stock prices for varying periods. The variation of accuracy with the forecasting period will be studied for different models.

The rest of the document has been structured into Literature review, Methods and Techniques employed for this study and a discussion on results and scope of further research.

II. Literature Review

Traditionally fundamental analysis played very important role in predicting the long-term trend of stock prices. The financial ratios produced from the fundamental analysis, gives intuition of the stock movement[1]. The process of using fundamental variables to make stock trading decisions begins with Benjamin Graham, as early as 1928. There has been numerous research works investigating and shaping the modern fundamental analysis. The fundamental analysis is essentially based on the key attributes of the enterprise for example earnings-to-price yield, Profit /Earnings ratios, Total debt, book price ratios etc.

While the fundamental analysis is helpful for long-term investors, the requirements of traders are not satisfied or often appear overlooked by these studies. Technical analysis is alternate discipline of financial markets study that fills the gap left by fundamental analysis. The temporal fluctuations in the stock price gives earning opportunities for the traders. Technical analysis is the field of science that deals with predicting the temporal fluctuations to support the traders make informed statistical decisions. The ability of predicting the prices measures the success of traders' portfolio. Technical analysis involves study of the stock prices based on pattern matching, measures of various technical indicators which are mathematical derivates. All the approaches of the technical analysis are dependent on three basic principles of technical analysis, namely: a) Prices move in trends b) Volume goes with the trend c) A trend, once established tends to persist. [1], [2]. The mathematical intuition behind the approaches of technical analysis makes the field an ideal use case to apply various statistical and machine learning based predictive models.

In a well-respected study by Neftci et al. [3] technical analysis was done on closing prices of gold and trade-bills. The results of the technical analysis were statistically evaluated. It was concluded that there is a significant relationship between the moving average and the close price of the stock.

Brick et al. in [4], by two technical trading rules ,moving averages and trend line are validated. It was concluded, that these technical rules result in statistically significant earnings.

Regardless of the random walk theory which states that the successive price changes in stock prices are independent and identically distributed random variables, numerous studies have been published with reasonable success in predicting the stock prices, using various machine learning techniques like classical time series, contemporary machine learning and deep learning methods. Broadly there are two type of systems exploited in predicting the stock prices - a) Statistical methods b) Machine learning and AI based deep learning methods.

Classical time series models are statistical methods employing linear processes as predicting techniques, such as the autoregressive integrated moving average model (ARIMA), the autoregressive conditional heteroscedasticity (ARCH) models.

In the work "Stock Price Prediction Using the ARIMA Model", [5] Adebiyi et al. presented extensive process of building stock price predictive model using the ARIMA. The study used one and half of decade data of two stocks from two different stock exchanges across the world New York Stock Exchange and Nigerian Stock exchange. The data was used to build ARIMA models. Results obtained revealed that the ARIMA model has a strong potential for short-term prediction and it was envisaged that the model can compete favourably with existing techniques for stock price prediction. Although the ARIMA model built is able to forecast the prices reasonably well, there is no comparative evaluation of the models. This creates a gap in statistically validating the performance of the models.

In the study," An Effective time series analysis for stock trend prediction Using ARIMA Model for Nifty Midcap-50" [6] Uma B et al. have analysed the 5 years' worth of data of the top four stocks from National Stock Exchange, India. ARIMA model has been proposed as a favourable model reasonably identifying the trends and able to forecast the prices. However, there is significant mean absolute percentage error in the forecasts and there is a very good scope of improving the model by employing one step forecasting.

Mondal et al. in their work [7] studied the generalizability of the ARIMA model by training the model using twenty-three months of data related to fifty-six different stocks from range of sectors from National Stock Exchange, India. The performance of the model based on size of data and generalizability of the models to different sectors has been studied. It was concluded that, ARIMA model performed well for the stocks which have seen relatively lower standard deviation. In addition, it was evidenced that the changes in accuracy for different sizes of data is not significant. There are some key gaps that have been identified in this literature. The auto ARIMA is not always guaranteed to converge to a perfect order of differentiation, regression and differencing of the time series. Investigation of root node of the time series, auto correlations and respective correlated residuals optimise the convergence of the model. This methodology will be implemented in the current study and manual Arima will be compared with Auto Arima.

In their very recent work [8], Kumar et al. compared the performance of contemporary machine learning classifiers in recommending 'Buy' and 'Sell' for range of stocks. Support Vector Machine (SVM), Random Forest, K-Nearest Neighbours (K-NN) and Naïve Bayes classifies have been compared. It was concluded that Random Forest algorithm outperforms the other algorithms. However, with minimum training data, Naïve Bayes classifier outperformed Random forest classifier. It was also identified that performance of the models increased with addition of technical indicators as features.

However, prediction systems based on statistical methods have their own limitations as they require more historical data to meet statistical assumptions for example identifying the cyclic trends in the data, other statistical attributes like stationarity and causality. In addition to this, most of statistical models are univariate in nature and therefore, additional features that can impact the stock price will remain transparent to the model. These characteristics inevitably impacts the short to long terms predictions. In addition to this problem, the models will fail to generalise on stocks belonging to a similar sector.

With the advancements in the hardware technology and the advent of deep learning, there have been increasing attempts to apply deep learning techniques to stock price prediction. Neural Networks (NNs) are inherently data-driven, adaptive and non -linear methods with few prior assumptions than the linear models. In [9] Abdul et al. have compared the performance of statistical model ARIMA and Artificial Neural Network (ANN). Although the error in the forecast was within acceptable levels for both AIRMA and ANN, it is understood that, there is a tremendous scope of improvement in this study. There are various variations of ANN model, which can be applied and optimised for enhancing the performance of the model. Intuitively the performance of ANN model is expected to be superior to that of ARIMA.

In the study, [10] Kara et al. compared two non-linear classification techniques viz SVM and ANN to predict the direction of stock movement in Istanbul Stock Exchange using a decade worth of trading data. It was observed that both the SVM and ANN have performed well in predicting the direction, with ANN performing marginally better at 75.74% accuracy as against 71.52% of SVM. One of the major factors that has boosted the model performance while controlling the variance, is adding various technical indicators in the data. These technical indicators played key role in predicting the movement of price.

In the recent work [11] Ugur et al. trained a 2 Dimensional Convoluted Neural Network (2D CNN) network using Exchange Traded Fund (ETF) data of New York stock exchange. The 14 years' worth of daily historic stock prices and respective technical features have been used to create a sliding window tensor to train 2D CNN. It has been concluded that the 2D CNN model has outperformed other classical predictive models with Mean Absolute Percentage Error (MAPE) of 72.9%.

In a notable comparative study [12] three neural network models - time delay, recurrent, and probabilistic neural networks are compared. It was concluded that the Recurrent Neural Network (RNN) showed the best performance among other models due to its inherent capability of incorporating temporal dimensions of the data as a result of internal recurrence. One of the key improvements that can be made in this study are scoping in family of RNN networks and optimising these networks.

In the study, [13] Lee et al. compared the performance of the Seasonal ARIMA (SARIMA) model with Back Propagation Neural Network(BPNN), while predicting the weekly and monthly averages of Korean Stock price index (KOSPI). The SARIMA model provided more accurate forecasts for the KOSPI than the BPNN model does. This relative superiority of the SARIMA model over the BPNN model is pronounced for the mid-range forecasting horizons. However, the difference in forecasting accuracies of the two models was not found to be statistically significant. In addition, it appears inappropriate to model the stock price as a seasonal component. The seasonality could be attributed to the impact of confounding variables for example, periodical announcements by the enterprise.

The recent work [14] Torres D G et al. employed the state-of-the-art Recurrent Neural Networks in multivariate time series forecasting scenario , to predict the future sequence. Two variants of RNN – Long Short-term memory (LSTM) and Gated Recurrent Unit (GRU) have been applied to predict the next day close price of the Bitcoin data. The performance of LSTM and GRU are compared with ARIMA and ARIMA Dynamic regression. It has been concluded that there is no considerable difference in terms of prediction accuracy between LSTM and GRU. However, both deep learning models outperformed ARIMA models, as one step forward forecasting technique is not used. The results of the model indicate symptoms of overfitting in case of LSTM and GRU. Regularising the models and optimising them can lead to better performing models. There have been recent studies employing evolutionary algorithms in optimizing the deep learning network parameters and this has reasonably enhanced the predicting capability [15], [16].

In the very recent work [17] Chou J et al. employed a non-linear modelling technique - Least Squares Support Vector Regression (LSSVR) to predict the next day close prices of stocks of Taiwanese construction companies. This methodology has achieved far greater accuracy in predicting the close price of the stocks compared with contemporary models. In the current study, the sliding window technique will be used for deep learning models. The time series models will be modelled on one step ahead methodology.

Motivated by the success of deep learning algorithms, considerable work has been done in exploring the hybrid models for predicting the stock price and associated price movements.

The study [18] Pai et al. proposed hybrid models in predicting the stock price using ARIMA and SVMs. 50-day historic data of ten different stocks have been used to predict the close price of respective stock using one step ahead forecasting technique. In this hybrid model, ARIMA model was used to estimate the close price of stock price. The residuals of the ARIMA model were then calculated and passed to SVMs model to predict the errors. These predicted errors were used to correct the ARIMA predictions. The hybrid model has dominated the single ARIMA and single SVMs performance.

There have also been some approaches to integrate qualitative information with deep learning techniques for stock market forecasting. Yoshihara et al. in the work [19] exploited the textual information as input variable and predicted market trends based on RNN model combined with restricted Boltzmann machine (RBM) to investigate the temporal effects of past events.

III. Methods and Techniques

To make reasonable comparison, the context of prediction can be divided into short-term and long-term prediction. The short-term prediction is about forecasting the stock price up to the next 2 days and long-term forecasting is to predict the price up to 5 days.

A. Data

The data comprises daily historic prise of Lloyds Bank Group share for 10 years, from 01 January 2009 until 31 December 2019.The daily prices include the Open , High , Low and Close price. The close price is by definition bound with in the range of the three prices. There are a total of 2857 data points. In addition to the Lloyds Banking Group share, the data comprising daily historic price for the last 10 years, for two additional shares is selected – Barclays Bank plc and Royal Bank of Scotland plc. This data is selected to evaluate the generalizability of the models.

Technical indicators which statistically describe the movement of stocks are found to be very useful features in predicting the future price [10]. These technical indicators are calculated based on the Lloyds stock price features viz Open, Close, High and Low. Table I shows the mathematical formulae of these indicators.

Deep learning and Time series models are supervised learning algorithms and as such, these models will need training data in the form of learning features (Independent variables) and ground truth label or regression outcome (Dependent variable). The data sourced for this study is a time series data and so, the data is converted in to supervised learning format using custom data generators.

TABLE I. TECHNICAL INDICATORS OF STOCK

Simple 10-day moving average	$\frac{(C_t + C_{t-1} + C_{t-2} + \cdots + C_{t-10})}{10}$
Weighted 10-day moving average	$\frac{(n * C_t + (n-1) * C_{t-1} + (n-2) * C_{t-2} + \cdots + 1 * C_{t-10})}{\frac{n * (n-1)}{2}}$
Momentum	$C_t - C_{t-n}$
Stochastic K%	$\left\{\frac{C_t - LL_{t-n}}{HH_{t-n} - LL_{t-n}}\right\} * 100$
Stochastic D%	$\frac{\sum_{i=0}^{n-1} K_{t-i}\%}{n}$
RSI(Relative Strength Index)	$100 - \frac{100}{1 + \frac{\frac{\sum_{i=0}^{n-1} Up_{t-i}}{n}}{\frac{\sum_{i=0}^{n-1} Dw_{t-i}}{n}}}$
MACD (Moving Average Convergence Divergence)	$MACD(n)_{t-1} + \frac{2}{n} + 1 * (DIFF_t - MACD(n)_{t-1})$
A/D Oscillator	$\frac{H_n - C_{t-1}}{H_n - L_n} * 100$
CCI (Commodity. Channel Index)	$\frac{M_t - SM_t}{0.015 D_t} * 100$
Larry William's R%	$\frac{H_n - C_t}{H_n - L_n} * 100$

C_t is the closing price , L_t the low price, H_t the high price , U_{pt} the upward price change, D_{wt} the downward price change at time t.

DIFF: $EMA(12)_t - EMA(26)_t$, EMA is exponential moving average, $EMA(K)_t$: $EMA(K)_{t-1} + \propto * (C_t - EMA(k)_{t-1}$; $\propto$ is the smoothing factor : 2/1+k where k is the time period of k day exponential moving average. $M_t = H_t + L_t + C_t/3$; $SM_t = (\sum_{i=1}^{n} M_{t-i+1})/n$;$D_t = (\sum_{i=1}^{n} |M_{t-i+1} - SM_t|)$.

B. Data Transformation

Each of the features used for training the models have different range leading to very wide feature space. Thus, the training of deep learning models is prone to swinging gradients. As part of the Data transformation step, the attributes of a dataset are normalized by scaling its values using the Min-Max normalisation technique. Mathematically the Min-max transformation can be represented as (1).

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{1}$$

C. Statistical Tests

The time series models like ARIMA have prior assumptions about the data, to ensure convergence to global minima, for example, the models assume that the time series should have no unit root is therefore stationary. Two types of statistical tests will be performed to satisfy these prior assumptions.

Augmented Dicky Fuller Test (ADFT) is used to identify the existence of the unit root for a time series component. If the time series has a unit root, then the time series under consideration can be categorised as a non-stationary series and the series will need to be converted to stationary series before applying statistical methods. If there is a unit root, then the series can be classified as non-stationary. However, lack of unit root doesn't make the series stationary [20].

Assume, a time series at any given time 't' can be represented as linear combination of value of the series at previous time step 't-1' and some error 'e' at the time t, then mathematically it can be represented as shown in (2).

$$y_t = \phi * y_{t-1} + e_t \tag{2}$$

if $\boldsymbol{\phi = 1}$, then the series has a unit root and will not be stationary. ADFT establishes a null hypothesis that the time series that is being tested will have unit root and is therefore non-stationary. Equation (3) represents mathematically the hypothesis test.

$$\Delta y_t = y_t - y_{t-1} = (\phi - 1) * y_{t-1} + e_t \tag{3}$$

Simple t-test will be done to check the probability of ϕ being equal to 1.

Granger Causality Test is used to identify the causal relationships between the feature vectors of a time series data. Assume y_t is the value of time series variable at time t and assuming, the time series is auto correlated, it can be represented as shown in (4).

$$y_t = \alpha_0 + \mu_t + \sum_{i=1}^{m} \alpha_i y_{t-i} \tag{3}$$

α_0 *is the time series constant*, α_i *is the linear* equation coefficients and μ_t *is the noise* in the time series.

Another time series x_t is said to be having granger causal relationship with time series y_t, if it can be linearly expressed as shown in (5).

$$y_t = \alpha_0 + \mu_t + \sum_{i=1}^{m} \alpha_i y_{t-1} + \sum_{j=1}^{m} \beta_j x_{t-j} \tag{5}$$

The null hypothesis of Granger causality test establishes that, two stationary time series components are statistically not causally related. At 95% confidence, if the p-value of the test is less than the significance level 0.05, the null hypothesis can be rejected. The pre-requisite of the Granger causality test is that the two series are stationary or co-integrated; otherwise the problem of 'spurious regression' might occur [21]. The Fig. 1 shows the summary results of ADF and Granger Causality test.

D. Modelling

The following time series and deep learning models are trained and evaluated-

- Time Series Models – Manual and Auto ARIMA, Vector Auto Regression.
- Deep learning Models – LSTM, GRU, CNN2D-LSTM and CNN1D.

1) ARIMA Models: ARIMA models forecasts the time series by modelling the predictor variable as a regressor. The time series is assumed to be a composition of three main components. The first component is Auto regressor (AR) which measures the correlation between an instance of the time series variable and the variable itself with a time lag. Equation (6) shows the AR component of the time series.

$$yt=c+\phi 1\ yt\text{-}1+\phi 2\ yt\text{-}2\ +\cdots+\phi p\ y\ t\text{-}p+\varepsilon t \qquad (6)$$

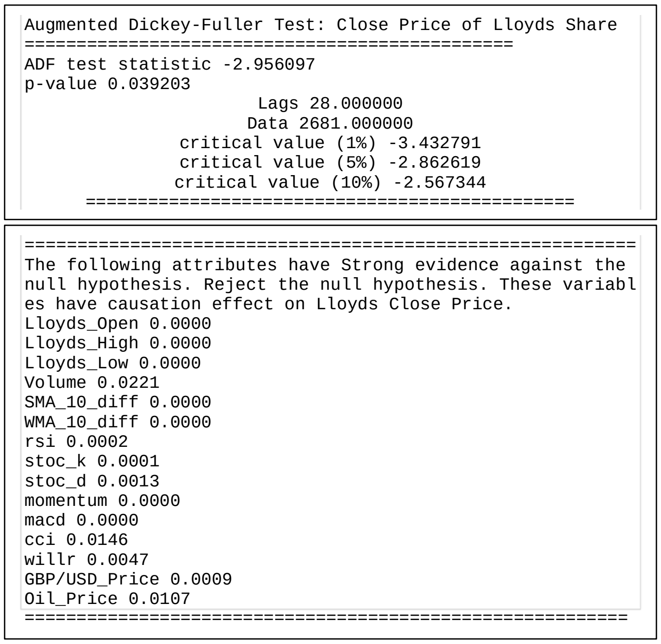
The above time series is also called the Auto regressive representation of the time series of order 'p'. $\varepsilon_{t.}$ is the error component in the time series. Just as linear regression, the algorithm learns the weights of the variables, while minimising the error.

The second component of the time series is the differencing which involves converting non-stationary series to a stationary series. A stationary series is a series which has constant mean and variance. The properties of the series are independent of the time. On the other hand, time series whose properties vary with time are called non-stationary series. Statistical tests, for example ADFT can be used to identify the stationarity of the time series. The process of differencing is finding the difference of the consecutive instance of variables. The differencing of the time series will stabilise the mean of the time series and therefore, the successive differencing will yield a stationary time series. The properties of a stationary time series viz. mean, variance and correlation help to accurately forecast the time series, as these properties do not vary with time. Thus, the stationarity is an important property of the time series for applying ARIMA model.

The third component of the time series is moving average (MA) component. The MA component represents the linear relationship of the predictor variable using the residuals of the forecast. Equation (7) shows the MA component of the time series.

$$Y_t=c+\varepsilon_t+\theta_1\ \varepsilon_{t\text{-}1}+\theta_2\ \varepsilon_{t-2}+\cdots+\theta_q\varepsilon_{t-q} \qquad (4)$$

ε is the residual of the forecast, technically called as 'white noise'. The above equation is called the moving average component of order 'q' of given time series. Just like the auto regression parameters, the model learns the weights 'θ' for optimising the errors in prediction.



```
Augmented Dickey-Fuller Test: Close Price of Lloyds Share
=============================================
ADF test statistic -2.956097
p-value 0.039203
                      Lags 28.000000
                     Data 2681.000000
               critical value (1%) -3.432791
               critical value (5%) -2.862619
              critical value (10%) -2.567344
      =============================================
```

```
========================================================
The following attributes have Strong evidence against the
null hypothesis. Reject the null hypothesis. These variabl
es have causation effect on Lloyds Close Price.
Lloyds_Open 0.0000
Lloyds_High 0.0000
Lloyds_Low 0.0000
Volume 0.0221
SMA_10_diff 0.0000
WMA_10_diff 0.0000
rsi 0.0002
stoc_k 0.0001
stoc_d 0.0013
momentum 0.0000
macd 0.0000
cci 0.0146
willr 0.0047
GBP/USD_Price 0.0009
Oil_Price 0.0107
========================================================
```

Fig. 1. Summary of Statistical Tests.

For manual ARIMA, the auto regression and moving average components of the time series are calculated manually using the auto correlation and partial auto correlation plots. There is an extensive research and a standard methodology proposed for calculating these orders [22]. These guidelines are followed to manually derive the AR and MA orders for forecasting the Lloyds Share close price. The concept of the auto ARIMA is much similar to the manual ARIMA. In Auto ARIMA, the order of the linear equation and the associated weights (Regressor, Integrated and Moving Average) are determined programmatically using the Hyndman-Khandakar algorithm [23].

The goodness of the models in Auto ARIMA is measured by Akaike Information Criteria (AIC) and Bayesian Information Criteria(BIC). AIC and BIC measures are used to estimate the likelihood of a model forecasting the future variable. These indicators are the measure of a good fit of a model. Equations (8) and (9) shows the mathematical formulae of these measures.

$$AIC = -\ 2\ \ln(L) + 2\ k \qquad (5)$$

$$BIC = -\ 2\ \ln(L) + 2\ \ln(N)\ k \qquad (6)$$

L is the value of likelihood; N is the number of observations and k is the number of estimates parameters.

2) Vector Auto Regression: Vector Auto Regression (VAR) is a statistical model used to capture the linear interdependencies among multiple time series. VAR models generalize the univariate autoregressive models (ARIMA models) by allowing more than one evolving variable [24].

VAR models require the features of time series variables to affect each other temporally. For Lloyds share close price prediction, there is a clear relation between the open, high and low price of the share. Mathematically, each variable is expressed as the linear combination of the variable itself at a time lag and the past values of the other variables at similar lags, as shown in (10) and (11).

$$y_{1,t}=c_1+\phi_{11,1}y_{1,t-1}+\phi_{12,1}y_{2,t-1}+e_{1,t} \quad (7)$$

$$y_{2,t}=c_2+\phi_{21,1}y_{1,t-1}+\phi_{22,1}y_{2,t-1}+e_{2,t} \quad (8)$$

The VAR model for 'n'th cointegrated variable can be represented as a linear combination of past values of 'n'th variable itself and the past values of all the other co-integrated variables. Algorithm learns the weights of the variables, while minimizing the error. Equation (12) shows the mathematical formulation of VAR.

$$Y_{n,t}=c_n+\phi_{n1,1}y_{1,t-1}+\phi_{n2,1}y_{2,t-1}+\phi_{n2,1}y_{2,t-1}+\ldots\ldots\ldots.+\phi_{nn,1}y_{n,t-1}+e_{n,t} \quad (12)$$

3) Recurrent Neural Networks (LSTM and GRU): Long short-term memory (LSTM) and Gated recurrent Unit (GRU) belong to class of artificial recurrent neural network (RNN) architectures.

Fig. 2 shows the basic architecture of the RNN [15]. In addition to the dense connections between the layers of the network, the RNN networks have additional feedback connections within neurons. Thus, each neuron of the network receives the input vector which is the output of activation function of previous neuron and the output of the time step (feedback connection), thus allowing the network to learn temporal changes in the data. The output of a particular neuron 'Y_t' at any time 't' can be represented mathematically as shown in (13).

$$Y_t = \Phi\,(W_x x_t + W_y y_{t-1} + b) \quad (9)$$

W_x are connection weights at time 't' with respect to feedforward input, W_y is the connection weights at time 't' with respect to feedback connection, b is the bias term. x_t is the feed forward input and y_{t-1} is the feedback input of the neurons. Φ is the activation function. The activation function will be normalised as part of optimisation of the performance of the network. One of the main disadvantages of the RNN is high susceptibility to vanishing and exploding gradients. Due to the number of weights involved and temporal connections within the neurons, the output of back propagation reduces drastically as the depth of network increases. Thus, the effect of back propagation fades away.

LSTM Networks are enhanced RNNs which works exactly in the same way as RNN, except that the neurons in the network are actually composite cells. Fig. 3 shows a common LSTM neuron, which is composed of an input gate, an output gate and a forget gate.

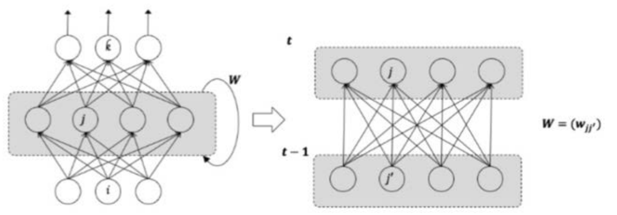


Fig. 2. Basic Architecture of RNN.

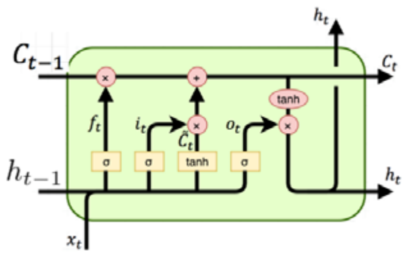


Fig. 3. LSTM Cell.

The input to each cell of LSTM at any time 't' comprises of three signals, a long-term memory C_{t-1},a short-term memory h_{t-1} and x_t [25] . f_t. represents the forget gate, i_i. represents the input gate and o_t represents the output gate.

The forget gate controls, what part of the long-term memory is erased. The forget gate does a multiplicate operation on the long-term state and logistic activation function of short-term signal at previous time step and input to the cell at current time step. Equation (14) represents the backpropagation of the forget gate.

$$f_t = \sigma(W_{xf}^T x_t + W_{hf}^T h_{t-1} + b_f) \quad (10)$$

The input gate controls, what values of the current time state can be added to the long-term memory. This gate is the main gate which takes the current time state(x_t) and short memory state from previous time step(h_{t-1}), applies the respective activation functions and adds the result to the long-term state. There are two activation functions one being the logistic function and the other one is a 'tanh' function. These functions are used to feed in the temporal state of the input to long term memory. Equation (15) and (16) shows the inputs of the input gate.

$$\overline{C_t} = tanh(W_{xc}^T x_t + W_{hc}^T h_{t-1} + b_c) \quad (11)$$

$$i_t = \sigma(W_{xi}^T x_t + W_{hi}^T h_{t-1} + b_i \quad (12)$$

The output gate controls what part of the long-term memory should be added to the output of the cell (h_t = y_t.). Equation (17) represents the net outcome of the output gate.

$$o_t = \sigma(W_{xo}^T x_t + W_{ho}^T h_{t-1} + b_o) \quad (13)$$

Summarizing the output of each gates and applying the additive and multiplicative operations, the final output of the LSTM cell is shown using equations.

$$C_t = (C_{t-1} \otimes f_t) \oplus (i_t \otimes \overline{C_t}) \quad (14)$$

$$h_t = o_t \otimes \tanh(C_t) \quad (15)$$

$W_{xf}^T, W_{xo}^T, W_{xi}^T$ *and* W_{xc}^T are the weights of the four gates within the cell for the connections to the given input *x*.

$W_{hf}^T, W_{ho}^T, W_{hi}^T$ *and* W_{hc}^T are the weights of the four gates within the cell for the connections to the given short-term state.

b_f, b_i, b_c, b_o are the bias terms for the four gates.

C_t and h_t are long- and short-term memories respectively.

As evident from the mathematical equations of the LSTM, the number of trainable parameters increases exponentially as the depth of layer increases, slowing down the training time. LSTMs are also prone to overfitting due to the activations at forget gate.

GRU networks on the other hand are the simplified version of the LSTM networks.

Fig. 4 shows the GRU network cell. The cell consists of input gate and a forget gate. Both the states, the long -term and short-term memory, at any time *‘t’* is merged into a single state h_t.

There is no output gate control and a single controller controls both input and forget gate. At any given time-step, the full state vector is output without any activation. Mathematically, the control equations are on similar lines to that of LSTM and represented as shown in (20), (21), (22) and (23).

$$z_t = \sigma(W_{xz}^T x_t + W_{hz}^T h_{t-1} + b_z) \tag{16}$$

$$r_t = \sigma(W_{xr}^T x_t + W_{hr}^T h_{t-1} + b_r) \tag{17}$$

$$\overline{h_t} = tanh(W_{xg}^T x_t + W_{hg}^T (r_t \otimes h_{t-1}) + b_g) \tag{18}$$

$$h_t = z_t \otimes h_{t-1} + (1 - z_t) \otimes \overline{h_{t-1}} \tag{19}$$

$W_{xz,}^T W_{xr}^T$ *and* W_{xg}^T are the weights of the gates within the cell for the connections to the given input *x*.

$W_{hz,}^T W_{hr}^T$ *and* W_{hg}^T are the weights of the gates within the cell for the connections to the given short-term state (temporal feedback).

b_z, b_r *and* b_g are the bias terms for the gates.

$\overline{h_t}$, h_t , z_t, *and* r_t are the output states at respective gates.

GRU has a smaller number of trainable parameters compared to the LSTM. Due to these reasons, the training time of GRU is much smaller and the network is less susceptible to vanishing gradients and overfitting, compared with LSTM. In the recent work, [26] it was observed that the GRU and LSTM performance is reasonably same. In this study, python Keras deep learning APIs for LSTM and GRU will be used for model training and optimisation.

4) Convolution Networks: Convolution Neural Network (CNN) is class of deep learning neural network which can accept the data in a multi dimension matrix form called tensor and learns the intrinsic dependencies of the data.

The most important building block of CNN is a convolution layer. The convolution layer works on the principle of mathematical function called ‘convolution’. Mathematically, convolution of two functions produces the third function, which explains how the shape of one function is modified by the second function.

$$f(t) * g(t) = \int_{-\infty}^{\infty} f(\tau)\, g(t-\tau) d\tau. \tag{20}$$

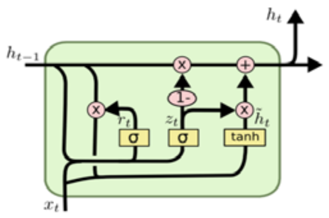


Fig. 4. GRU Cell.

Equation (24) shows two functions $f(t), g(t)$ which are convoluted. The convolution involves bounded integral of the product of two functions represented as a function of a temporal change τ.The resultant product is the convolution of the two functions. CNN applies these convolutions on set of pixels rather than each pixel to extract the abstract features from given data. The phenomenon of extracting the abstract features is called pooling.

The features extracted from each layer of the CNN is represented as feature map [27]. Multiple CNN layers extract the feature maps successively and the intricate dependencies within the features is learned by the model. Fig. 5 shows the architecture of CNN.

1D CNN is the extension of the CNN architecture, where the network can accept one dimensional sequence of the data and feature extraction is done using the convolutions. The extracted abstract features are used for forecasting the output. Therefore, this architecture of CNN can be used for time series forecasting problems. Several 1d convolution and pooling layers, stacked each other, makes the network powerful to extract the hidden dependencies within the temporal data. A filter of size *(1 x m)* is convoluted on a time step of size *(1 x n)*. Each stride of the filter extracts abstract feature from the time step, creating a feature map. Such feature maps are pooled through successive convoluted layers for forecasting the time series.

1D CNN stacked LSTM network is a hybrid network used to forecast the temporal data. Fig. 6 shows 1D CNN+ LSTM network. The network architecture consists of a convolution layer, which accepts the input as a tensor and performs convolutions using appropriate strides. The convoluted feature map is fed to LSTM. Deep LSTM network learns the convoluted feature maps and forecasts the time series. Therefore, 1D CNN is used as feature extractor and LSTM is used as sequence predictor.

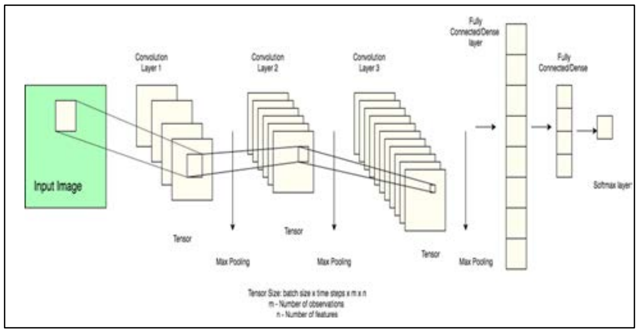


Fig. 5. General CNN Architecture.

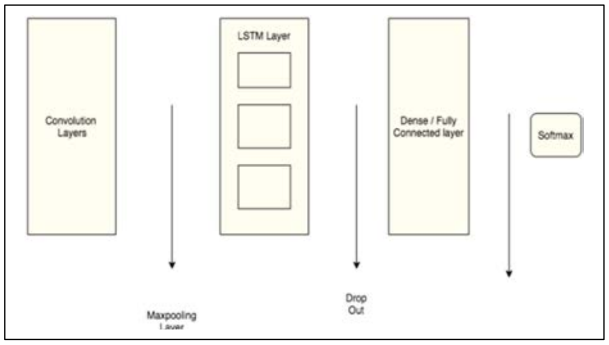


Fig. 6. CNN and LSTM Network Architecture.

5) Performance optimisation: All the deep learning networks mentioned above have respective hyper parameters. These hyper parameters will be optimised using Genetic algorithm [15], [16] and Tree-structured Parzen Estimator Approach (TPE Approach) [28].

Genetic algorithm will be used to identify the optimal number of time steps that algorithms use for learning the sequence. This is the temporal dimension that the model should consider for forecasting. TPE optimization is used for optimising the rest of hyper parameters viz. batch size of the inputs, which is the number of observations that network will learn for adjusting the weights following feedforward and back propagation cycles, number of neurons with in each layer of the network, the activation functions within each dense layers, the size of drop out, learning parameter and learning optimizer function.

Mean Absolute Percentage Error (MAPE) and Mean Absolute Error (MAE) will be used to evaluate and select the models. Percentage errors have the advantage of being scale-independent, and so are frequently used to compare forecast performance across different data sets. These errors put a heavier penalty on positive errors than on negative errors.

Assume, y_{pred} is the predicted value of a time series and y_{actual} is the actual value. Then, MAPE and MAE are mathematically represented as

$$\text{MAPE} = \text{mean}\left\{\frac{100 * ||(y_{actual} - y_{pred})||}{y_{actual}}\right\} \tag{21}$$

$$\text{MAE} = \text{mean}\left(\left|\left|y_{actual} - y_{pred}\right|\right|\right) \tag{22}$$

The best performing models from time series and deep learning classes will be compared based on following characteristics:

- The measure of error (MAPE) in forecasting the short-term prices and long-term errors.
- The measure of error (MAE) in forecasting the short-term prices and long-term errors.
- The change in the measure of error when exogenous variables are added to the model.
- The ability to generalize and perform on peer stocks.

IV. Results

As part of this study, the performance of the models while forecasting close price of Lloyds share is measured. The length of forecasting period is limited to 5 days, to avoid the impact of volatility due to sudden political or economic shocks within the share market. However, to study the general model performance on the test data and to support residual analysis, the size of test data is set to higher value than maximum forecasting period.

The total observations are split in 90%:10% ratio as training and test data sets respectively. This ratio yields 2439 as training observations and 271 as test observations.

The test observations are further split into two equal sets. These sets are used for cross validation and out of sample tests. This methodology allows having equal test size data sets for time series and deep learning models. Table II shows the data set sizes.

For each of the time series and deep learning models, metrics are recorded against each characteristic, and the conclusions are drawn based on these. Table III shows the performance of the time series models.

In terms of execution time, all the models performed roughly on same scale, with VAR marginally running for longer duration. While ARIMA based models builds the linear relationship between stock price and itself at different lags, the VAR model builds linear equation based on intrinsic relationship between each of the explanatory features in addition to the dependencies within the features at different time lags. This results in higher execution time compared to other linear models.

Based on the MAE and MAPE metrics, ARIMA models stands out. The manual ARIMA has performed marginally better when compared with auto ARIMA. In case of auto ARIMA, the orders of three components of ARIMA model viz AR, I and MA are calculated programmatically as (2,1,0). However, based on the heuristic methodology, as part of manual ARIMA modelling, the order (1,1,0) has been used to train the model. It is evident that the over differencing of the time series resulted in loss of performance, in case of auto Arima.

Table IV shows the performance of the deep learning models. The training time plays a crucial role in predicting the close price of the stock. In commercial context, a very high execution time, meant that there is loss of opportunity and the model predictions might not be relevant to the day and time of trade. The training time of CNN1D+LSTM is exceptionally high nearing 22hrs on an Nvidia Tesla K80, 4 core CPU. Therefore, the prediction of the model loses business justification and is practically not feasible. Same is the case with LSTM model with training time close to 20hrs.

TABLE II. TECHNICAL INDICATORS OF STOCK

Complete Dataset	2710
Training Dataset	2439
Cross Validation Dataset	136
Test Data Set	135

TABLE III. METRICS OF TIME SERIES MODEL

Parameter	Manual ARIMA	Auto ARIMA	VAR
Training Time	0:00:03	0:00:47	0:01:23
Training MAE	21.029	1.029	1.032
Training MAPE	31.708	1.718	1.718
Test MAE (2 days)	1.684	0.491	0.565
Test MAPE (2 days)	2.863	0.884	1.016
Test MAE (5 days)	1.194	0.590	0.657
Test MAPE (5 days)	2.039	1.058	1.177

TABLE IV. METRICS OF DEEP LEARNING MODEL

Parameter	LSTM	CNN1D + LSTM	GRU	CNN1D
Training parameters	103,937	634,049	385,409	18,465
Training Time*	19:36:42	21:52:13	11:45:47	7:34:37
Training MAE^τ	0.0155	0.0007	0.0011	0.0005
Test MAE (2 days)	0.008	0.007	0.0040	0.0110
Test MAPE (2 days)	2.110	1.901	0.9850	2.907
Test MAE (5 days)	0.005	0.004	0.0040	0.007
Test MAPE (5 days)	1.413	1.021	1.1760	1.807

* The training time includes the time taken to optimise the network using Tree Parzen Estimator random search and genetic algorithm-based optimisation techniques. The training time depends on the total number of parameters used for training.

τ The networks are trained using multiple epochs to reduce the bias. Therefore, the training MAE is the average value of all the epochs.

Considering the training and test errors, CNN1D has recorded higher test error compared with mean training error. This is an overfitting characteristic and the model is highly prone to bias and variance. In terms of MAE and MAPE the GRU has performed marginally better compared with CNN1.

Based on the out of sample evaluation of the models, Manual ARIMA and GRU models are selected as favourable forecasting models.

The RNN – GRU and manual ARIMA models are trained and evaluated on RBS and Barclay's stock data. Table V shows the evaluation metrics of RNN—GRU and ARIMA models. Deep learning models outperform the ARIMA model in predicting the short-term prices. ARIMA model, on the other hand is able to generalise the stock price towards the tail end of the forecasting period with no significant betterment over the deep learning models.

TABLE V. DEEP LEARNING VS TIME SERIES MODEL

Stock	Model	MAE_1_Day	MAE_5_Day
LBG	RNN GRU	0.004	0.004
LBG	Manual ARIMA	1.402	1.004
Barclays	RNN GRU	0.027	0.014
Barclays	Manual ARIMA	3.66	2.783
RBS	RNN GRU	0.038	0.028
RBS	Manual ARIMA	5.736	5.552

V. CONCLUSION

Table VI shows the comparison of residuals of the optimised GRU with that of contemporary research papers. The optimised deep learning model has shown superior performance in predicting the close price and is highly viable model from commercial context. Deep learning models outperformed time series models for both short- and long-term forecasts. Fig. 7 shows the performance of the GRU and Manual ARIMA models in forecasting the long term and short-term stock price.

TABLE VI. COMPARISON OF OPTIMISED MODEL AND PREVIOUSLY PUBLISHED MODELS

Authors	Algorithm/Models	Metric and Error	Optimised RNN – GRU trained as part of this study
Haider Khan Z, Alin A S, Hussain T	ANN	MAE (for 1-day forecast)- **0.0174**	MAE (for 1-day forecast)- **0.002**
Torres Douglas, Qiu Hongliang	RNN - LSTM and GRU	RMSE (for 1-day forecast) ARIMA - **147.6** LSTM - **518.6** GRU - **396.4**	RMSE for 1-day forecast – **0.004**
Chung Hyejung, Shin Kyung Shik	GA Optimised LSTM	MAPE (1-day forecast) – **0.91**	MAPE for 1-day forecast – **0.592**
Patel Jigar, Shah Sahil, Thakkar Priyank, Kotecha K	Hybrid Models - Fusion of SVR, ANN and RF	MAPE (5-day forecast) SVR–ANN - **11.2** SVR–SVR - **2.41** SVR–RF - **11.31**	MAPE for 5-day forecast – **1.176**
Lee Kyungjoo, Jin John Jongdae	SARIMA and ANN	MAE (7 – day forecast) ANN - **1.110** SARIMA - **1.927** ANN-SARIMA – **0.742**	MAE for 7-day forecast – **0.004**

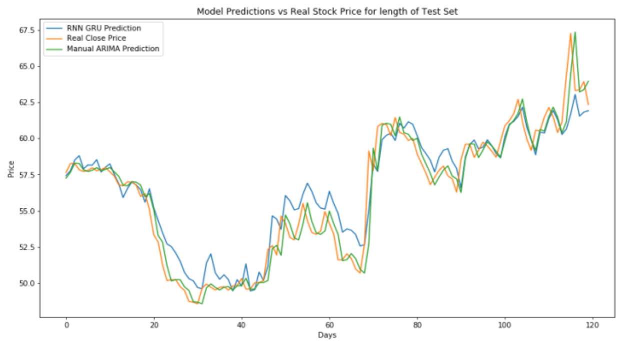


Fig. 7. Performance of ARIMA and GRU Models.

The approach and the experiments adopted in this study, reasonably answered the objectives of the research viz. comparing the performance of Time Series and Deep learning models, study the generalizability of the models and behaviour of the models for predicting long term forecasts. The generalizability of the deep learning models is superior to that of the time series models. This is particularly true for banking sector, in which the stocks appear to be correlated with confounded variables. With the increase in the length of forecast, the deep learning models can generalise the models well compared with time series models. However, there is a definitive length of the forecast during which this behaviour is observed. Beyond this forecast length, the volatility in the market outweighs the model behaviour.

However, there are some areas where, further research can be applied. Vector Auto Regression which can regress the time series data by modelling the series as a linear combination of the series itself at different lags and the respective cointegrating time series, has performed poorly compared with univariate time series models. This can be attributed to variance in the unit root of the cointegrating time series viz. technical and economic indicator. Further research can be done to understand the nuances of VAR and explore the limitations and scope of improvements. In addition, the sliding window technique used for training the deep learning models can be applied to linear time series models viz. ARIMA and VAR to construct the temporal dimension to enhance the performance. There has been appreciable research in the area of reinforcement learning for forecasting the movement of stock price. Research in measuring and characterising the reinforcement algorithms by comparing them with deep learning algorithms will greatly help the knowledge base and commercial applications.

Acknowledgment

I would like to thank my mentor Dr.Manoj Jayabalan, depart of Applied Mathematics, Liverpool John Moors University, London, United Kingdom, M.Jayabalan@ljmu.ac.uk for the constant guidance, encouragement, valuable feedback and evaluation of this study.

References

[1] B. Vanstone and G. Finnie, "An empirical methodology for developing stockmarket trading systems using artificial neural networks," Expert Syst. Appl., vol. 36, no. 3 PART 2, pp. 6668–6680, 2009, doi: 10.1016/j.eswa.2008.08.019.

[2] M. J. Pring, Technical Analysis Explained. 2014.

[3] S. N. Neftci, "Naive Trading Rules in Financial Markets and Wiener-Kolmogorov Prediction Theory: A Study of 'Technical Analysis,'" J. Bus., vol. 64, no. 4, p. 549, 1991, doi: 10.1086/296551.

[4] W. Brock and J. Lakonishok, "American Finance Association Simple Technical Trading Rules and the Stochastic Properties of Stock Returns Author (s): William Brock , Josef Lakonishok and Blake LeBaron Source : The Journal of Finance , Vol . 47 , No . 5 (Dec ., 1992), pp . 1731-1764," J. Finance, vol. 47, no. 5, pp. 1731–1764, 1992.

[5] A. A. Adebiyi, A. O. Adewumi, and C. K. Ayo, "Stock price prediction using the ARIMA model," Proc. - UKSim-AMSS 16th Int. Conf. Comput. Model. Simulation, UKSim 2014, no. June, pp. 106–112, 2014, doi: 10.1109/UKSim.2014.67.

[6] Uma B, S. D, and A. P, "An Effective Time Series Analysis for Stock Trend Prediction Using ARIMA Model for Nifty Midcap-50," Int. J. Data Min. Knowl. Manag. Process, vol. 3, no. 1, pp. 65–78, 2013, doi: 10.5121/ijdkp.2013.3106.

[7] P. Mondal, L. Shit, and S. Goswami, "Study of Effectiveness of Time Series Modeling (Arima) in Forecasting Stock Prices," Int. J. Comput. Sci. Eng. Appl., vol. 4, no. 2, pp. 13–29, 2014, doi: 10.5121/ijcsea.2014.4202.

[8] I. Kumar, K. Dogra, C. Utreja, and P. Yadav, "A Comparative Study of Supervised Machine Learning Algorithms for Stock Market Trend Prediction," Proc. Int. Conf. Inven. Commun. Comput. Technol. ICICCT 2018, no. Icicct, pp. 1003–1007, 2018, doi: 10.1109/ICICCT.2018.8473214.

[9] S. A. Abdul-Wahab and S. M. Al-Alawi, Comparison of ARIMA and Artificial Neural Networks Models for Stock Price Prediction Ayodele, vol. 17, no. 3. 2014, pp. 219–228.

[10] Y. Kara, M. Acar Boyacioglu, and Ö. K. Baykan, "Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange," Expert Syst. Appl., vol. 38, no. 5, pp. 5311–5319, 2011, doi: 10.1016/j.eswa.2010.10.027.

[11] M. Ugur Gudelek, S. Arda Boluk, and A. Murat Ozbayoglu, "A deep learning based stock trading model with 2-D CNN trend detection," 2017 IEEE Symp. Ser. Comput. Intell. SSCI 2017 - Proc., vol. 2018-Janua, no. November, pp. 1–8, 2018, doi: 10.1109/SSCI.2017.8285188.

[12] E. W. Saad, D. V. Prokhorov, and D. C. Wunsch, "Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks," IEEE Trans. Neural Networks, vol. 9, no. 6, pp. 1456–1470, 1998, doi: 10.1109/72.728395.

[13] K. Lee and J. J. Jin, "Neural Network Model Vs. Sarima Model in Forecasting Korean Stock Price Index (Kospi)," Issues Inf. Syst., vol. 8, no. 2, pp. 372–378, 2007.

[14] D. G. Torres and H. Qiu, "Applying Recurrent Neural Networks for Multivariate Time Series Forecasting of Volatile Financial Data," pp. 1–10, 2018.

[15] H. Chung and K. S. Shin, "Genetic algorithm-optimized long short-term memory network for stock market prediction," Sustain., vol. 10, no. 10, 2018, doi: 10.3390/su10103765.

[16] G. G. Szpiro, "Forecasting chaotic time series with genetic algorithms," Phys. Rev. E - Stat. Physics, Plasmas, Fluids, Relat. Interdiscip. Top., vol. 55, no. 3 SUPPL. A, pp. 2557–2568, 1997, doi: 10.1103/physreve.55.2557.

[17] J. Chou, "Forward Forecast of Stock Price Using," IEEE Trans. Ind. Informatics, vol. 14, no. 7, pp. 3132–3142, 2018, doi: 10.1109/TII.2018.2794389.

[18] P. F. Pai and C. S. Lin, "A hybrid ARIMA and support vector machines model in stock price forecasting," Omega, vol. 33, no. 6, pp. 497–505, 2005, doi: 10.1016/j.omega.2004.07.024.

[19] A. Yoshihara, K. Fujikawa, K. Seki, and K. Uehara, "Predicting the Trend of the Stock Market by Recurrent Deep Neural Networks," Pacific Rim Int. Conf. Artif. Intell., pp. 1–11, 2014.

[20] A. Pal and P. Prakash, Practical Time Series Analysis. 2017.

[21] X. Wang, "A Granger causality test of the causal relationship between the number of editorial board members and the scientific output of

universities in the field of chemistry," Curr. Sci., vol. 116, no. 1, pp. 35–39, 2019, doi: 10.18520/cs/v116/i1/35-39.

[22] R. Nau, "Summary of rules for identifying ARIMA Models," Duke University, 2019. [Online]. Available: http://people.duke.edu/~rnau/arimrule.htm.

[23] Rob J. Hyndman and Yeasmin Khandakar, "Automatic Time Series Forecasting: The forecast Package for R," J. Stat. Softw., vol. 27, no. 3, p. 22, 2008.

[24] Rob J Hyndman and George Athanasopoulos, Forecasting: Principles and Practice, 2nd ed. Sydney, Australia: OTEXTS, 2018.

[25] Aurelien Geron, Machine -Learning with Scikit-Learn and TensorFlow, First Edit. California: O'REILLY, 2017.

[26] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A Search Space Odyssey," IEEE Trans. Neural Networks Learn. Syst., vol. 28, no. 10, pp. 2222–2232, 2017, doi: 10.1109/TNNLS.2016.2582924.

[27] A. Chernodub and D. Nowicki, "Orthogonal permutation linear unit activation function (OPLU)," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 9887 LNCS, no. July 2018, pp. 533–534, 2016, doi: 10.1007/978-3-319-44781-0.

[28] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," Adv. Neural Inf. Process. Syst. 24 25th Annu. Conf. Neural Inf. Process. Syst. 2011, NIPS 2011, pp. 1–9, 2011.