Financial Data Time Series Forecasting Using Neural Networks and a Comparative Study

Ritvik Khandelwal
Electronics and Telecommunication
Dwarkadas J. Sanghvi College of Engineering
Mumbai, India
ritvik02.kh@gmail.com

Prasham Marfatia
Electronics and Telecommunication
Dwarkadas J. Sanghvi College of Engineering
Mumbai, India
prasham21@gmail.com

Shubham Shah
Computer Science
Shah and Anchor Kutchhi Engineering College
Mumbai, India
shubham.champ12@gmail.com

Varun Joshi
Computer Science
Mukesh Patel School of Technology Management and Engineering
Mumbai, India
joshivp@gmail.com

Pratik Kamath
Computer Science
Mukesh Patel School of Technology Management and Engineering
Mumbai, India
pratikkamath2000@gmail.com

Kshitij Chavan
Computer Science
Mukesh Patel School of Technology Management and Engineering
Mumbai, India
kshitijchavan2000@gmail.com

Abstract: **Algorithms that are based on Machine learning & Deep learning are new methods to solve time series prediction challenges. These methods have been proved to yield better results than the traditional regression models. A study concluded that artificial Recurrent Neural Networks such as Long Short Term Memory and Autoregressive Integrated Moving Average models are top models for this application. This research reports analysis and comparison between neural networks such as Convolution Neural Network, Long Short Term Memory, Bidirectional LSTM and Autoregressive Integrated Moving Average models. Comparative analysis of the above models shows that the BiLSTM performs better than ARIMA, CNN and LSTM models.**

Keywords – LSTM, ARIMA, CNN, Time Series Analysis

I. INTRODUCTION

Due to the obvious high volatility and unpredictability of financial time series data, forecasting commercial time series data is a difficult and time-consuming undertaking. Therefore, improving the accuracy of predictions is necessary. In the financial and academic worlds, stock market forecasting is an important subject of research. Technical and fundamental data can be used to forecast stock trends. The term "technical analysis" refers to the process of predicting stock developments entirely based on historical data. Technical analysis is a part of stock trading in which statistical trends and charts are studied from trading activity, such as price action and volume to evaluate and select the best trading opportunities.

Among the most famous and commonly used time series data analytics and forecasting is the auto-regressive integrated moving average. This linear regression-based model has gotten better over time, and as a result, many various types of this model have formed, such as SARIMA and ARIMAX. For short-term trend forecasting, these models work brilliantly, however for long-term trend forecasting, they fail miserably.

In Artificial Intelligence based data analysis, machine learning, specifically, deep learning-based techniques are upcoming techniques. These Artificial intelligence and learning-based technologies boost data analytics to a new level, with models that really are data-driven instead of just model-driven. The best learning model for the underlying application can be trained. Convolution based neural networks, for example, are well-suited to image recognition challenges, but recurrent neural networks are more suited to modelling problems like time series data and analysis. LSTM models are a specific category of Recurrent neural network that represents the dependency between longer input data and output data. The variation of LSTM is Bidirectional LSTM which saves the data in backward direction.

These RNN, also known as feedback models, learn from past data by utilizing multiple gates in their network architecture to remember past data and then construct a projected model based on both historical and present data.

II. RELATED WORKS

Time series modelling and dynamic modelling is an interesting area of research with numerous applications in management, marketing, computer science, and money. The goal of the analysis is to look at the course of time series observations and create a model to understand the data structure and forecast future data. Due to the value of time series analysis it is critical to establish an effective forecasting system in many disciplines of applied sciences. In this research a number of models is compared for time series forecasting.

Traditional time series analysis and forecasting methods depends on (ARIMA) Autoregressive Integrated Moving Average, its various variants, including (SARIMA) Seasonal ARIMA and ARIMA with additional factors. [1]. For a long time, these methods have been used to model time series problems. [2] [3]. These average based methods function very well, although they are flawed in several ways.:

These models are unable to describe data with nonlinear connections between parameters, since they are both regression based solutions to the issue.

When conducting statistical tests, there are some assumptions about data which must be maintained in order to have a relevant model.

Nevertheless they are accurate and efficient for short term trend predictions ; they are inaccurate for long-term forecasting.

Machine learning & deep learning techniques have created new possibilities for analyzing data from a time series. To estimate the S&P 500, Kraus. used a variety of forecasting methods, including random forests , boosted trees, and algorithm based on deep learning. Training and testing of the neural network is hard, according to Krauss et al [4]. Lee & Yoo [5] proposed an RNN method for forecasting stocks returns on investment. The intention was to develop portfolios

by tweaking the RNN's internal layers to adjust the return threshold levels. Fischera et al. [6] have done similar work for financial data prediction. This paper uses CNN and LSTM model for predicting gold price data forecasting. In this research using zCNN, RNN, LSTM and model stock price prediction is performed [8]

This paper uses combination of CNN and LSTM techniques which gives the model ability to extract the useful knowledge and identify short-long term dependencies from the time series data. [7] This paper is based on the writer's earlier research, which evaluated the accuracy of ARIMA method and LSTM-based models in terms of predicting economic and parameter tuning , financial time series. [9]

This paper does an additional step in performing comparative analysis within LSTM, CNN, ARIMA and BI-LSTM, it is unclear whether bidirectional training Bi-LSTM learning time series data can benefiit in prediction of finacnial time series data.

III. Algorithms

A. Reccurent Neural Network

Recurrent neural networks are a category of Feed Forward neural network that can deal with variable-length sequence inputs. These networks can handle sequential inputs, with exception of traditional feed forward network, which are unable to handle successive input and require all the input and outputs to be not dependent of one another. RNN models have gates for storing and utilizing sequential information from prior inputs. Recurrent hidden states are a form of RNN memory which enables the RNNs to forecast what input will be presented next in a sequence of provided input data.

For arbitrarily lengthy sequences, RNNs can theoretically utilize prior sequential information. In reality, however, the duration of sequential data has limitation to only a few steps due to RNN memory constraints. Assuming $x = (x_1, x_2, x_3...., x_j)$ denotes a series of varying lengths j and h_j denotes at time j the RNN memory, an RNN refreshes the memory knowledge using:

$$hj = \sigma (W_X x_j + W_j h_{j-1} + b_j) \quad (1)$$

W_X and W_j are weight matrix, and b_j is bias which is constant, and is a nonlinear function. example, the logistic, hyperbolic tangent function, sigmoid or the rectified linear unit.

One input to many output, multiple input to many output, many input to one output are all possible configurations for RNNs. Only RNNs that provide one output are considered in this study.

$y = (y_1, y_2, y_3,..., y_j)$ is the chance of happening of the following event. While the previous inputs are given, an element of a sequence is created. The probability of a sequence can be further seperated into as described:

$$p(x_1 , x_2 .., x_j) = p (x_1) p (x_2 | x_1) p (x_3 |x_1, x_2)... p(x_j |x_1, ...x_{t-2}, x_{j-1}) \quad (2)$$

every distribution of probability of condition modelled as follows:

$$p(x_t | x_1 , x_2 ...x_{t-1}) = \sigma(h_i) \quad (3)$$

where h_i is obtained using equation number 1

Vanishing gradients are one of the most common RNN problems, when information about the input / gradient travels through several layers before diminishing and dying out when it reaches the beginning or end of the layer. Because of this issue, RNNs will have a difficult time capturing long-term dependencies, making RNN training extremely difficult. Another concern with RNNs is a phenomenon known as "exploding gradients," which occurs when information or information about an input flows through several layers, gathering and resulting in such a large gradient whenever it reaches the beginning or ending layer. RNNs are difficult to train because of this issue.

When a function's output is described mathematically or computed as the partial derivative of the function's output regarding the inputs, the gradient serves as a gauge for how often a function's output shifts in reaction to variations in its inputs. Under these conditions, The RNN training method values the weight matrix at a lower degree., which is utilized in the RNN training phase, and therefore the RNN model ceases to be able to learn. The training technique, on the other hand, provides bigger values to the weight matrix for no apparent reason in the new gradients problem. The gradients can be truncated or compressed to solve this problem. [9]

B. Autoregressive Integrated Moving Average Models (ARIMA)

ARIMA is a meaningful abbreviation since it represents the model's key features: AR (Auto regression), which relies on a relationship between previous and present observations; This is based on a correlation between recent and prior observations:
I - Integrated: comparing observations to see when events occurred steadily in the time series data; Moving Average (MA): the lags of the moving average model's forecast mistakes.
As a set of parameters, these elements are incorporated in ARIMA models. (p,d,q) is the standard symbol for the ARIMA model, with p denoting the count of observations which lags, d denoting degree which is used differencing, and q denoting the moving average window size

$$Y_j = \theta_0 + \phi_1 y_j - 1 + ... + \phi_p y_{j-k} + \varepsilon_t - \theta_1 \varepsilon_{j-1} - ... - \theta_q \varepsilon_{t-q} \quad (4)$$

Y_j denotes the current measured observation at time j; j indicated the random mistake at time j; I and j denote coefficients: p and q, respectively, denote the autoregressive and moving average particular parameters. Box and Jenkins apply a modification to the series to make it stationary if the ARMA process is dynamic and non-stationary, resulting in the ARIMA model. This is accomplished by substituting the calculated values y_j with the outputs of a recursively differencing procedure s $\nabla^d y_j$, where d is the count of instances the procedure has been conducted. The 1st order of differencing can be shown as below:

$$\nabla^d y_j = \nabla^{d-1} y_j - \nabla^{d-1} y_{j-1}. \quad (5)$$

C. Long Short-Term Memory Model

As previously noted, RNN have difficulty understanding long term dependencies. Long Short Term Memory are the RNN extensions that can solve the vanishing gradient problem with ease. The RNNs' memory is effectively extended by the LSTM models, allowing them to remember and learn long-term input dependencies. This memory feature enables them to recall knowledge for longer periods of time, allowing them to view, edit, and delete the data from memories. Long Short Term Memory is referred to as the gated cell, with word gate which refers to the capacity to choose whether to save or discard memory data. LSTM model takes key features from inputs and then saves them for later use. The information's weight values assigned throughout the training phase are used to determine if it need to be eliminated or preserved.

LSTM models typically have three gates: an input gate, a forget gate, and an output gate. First gate decides whether current information will be kept or deleted, the input gate describes how many latest information will be summed to the data memory, and the final gate describes whether the cell's current observation contributes to the output.

I) Forget Gate: The information decision is required to be erased from Long Short Term Model data is commonly made using a sigmoid function. This choice is primarily based on the values of h_{j-1} and x_t. This gate's output is is f_j, a number ranges from 0 and 1, with 0 denoting complete removal of the learned value and 1 denoting preservation of the entire value. This result is calculated as follows:

$$F_j = \sigma (W_{fh} [h_{j-1}], W_{fx} [x_j], b_f) \quad (6)$$

where the b_f is the bias value and is a constant.

II) Input Gate: The input gate is the second type of gate. This gate determines whether or not fresh data is stored in the LSTM memory.

The Sigmoid and Tanh Layers The tanh layer handles what values must be changed in which layer, while the sigmoid layer generates a new vector. LSTM memory child values. The outputs of these two levels are calculated using the following formula:

$$I_j = \sigma(W_{ih} [h_{j-1}], W_{ix} [x_j], b_i) \quad (7)$$

$$C_j = \tanh(W_{ch} [h_{j-1}], W_{cx} [x_j], b_c) \quad (8)$$

C_j provides a vector of newest child values to be inserted into the memory, as well as whether or not the value should be changed. Current value is obscured by the present rate using F gate layer by multiplying the oldest value (i.e., c_{j1}) and then increasing the value of the new candidate to c_j, The following equation represents as follow:

$$C_j = f_j * c_{j-1} + i_j * c_j \quad (9)$$

in where fj is the outcome of the gate, in which is a value ranging between 0 and 1, with 0 meaning that the data has been entirely lost and 1 suggesting that the value has been fully retained.

The output gate is the third component. The sigmoid layer is responsible for determining which part of LSSTM memory is used to generate the output. It is therefore necessary to convert the values among 1 and 0 using the non-linear tanh function. Ultimately, the output is multiplied by the result of a sigmoid stage to arrive at the final result. The following equation represents the formulae that are used to calculate the final result:

$$O_j = \sigma(W_{oh} [h_{j-1}], W_{ox} [x_j], b_o) \quad (10)$$

$$H_j = o_j * \tanh(c_j) \quad (11)$$

where o_j is the value of output and h_j is its symbol in a range of 1 to 1.

D. Deep Bidirectional LSTM (BiLSTM)

Bidirectional LSTM are a kind of LSTM which can be used to improve model metric in problems involving sequence categorization. In cases where all input time steps sequence are known, bidirectional LSTM train two of one LSTM on the input sequence. The first is based on the original input sequences, while the input second is based on a reversed duplicate of the first. This can provide additional meaning for the networks, allow to understand the problem more quickly and thoroughly. A bidirectional LSTM is a model for processing sequences that is made up of two LSTMs: one that inputs information and one that outputs information. BiLSTM enhances the data accessible to the network, giving the algorithm with a larger context in which to function.

Deep bidirectional LSTM extends the original LSTM model in which the input data is processed by two LSTM. bidirectional LSTM are an extension of the LSTM model previously presented forward layer. In the second round, The input sequence is reversed and then fed into the LSTM model. LSTM enhances the learning of long term interdependencies as a result the mode's accuracy when used twice. [10]

E. Convolutional neural network

The CNN, a type of artificial neural networks that has proven prominent in many computers vision tasks, is gaining attention across a wide range of disciplines, notably forecasting, and is expected to continue to do so. CNNs are intended to acquire spatial hierarchies of features dynamically and seamlessly via backpropagation, and they use a variety of building blocks, even with layers like as convolution, pooling, and complete connection, to do this. It is the purpose of this section to provide an overview of the fundamental principles of CNN and their application to this report.

Local connectivity was a driving force for the development of convolutional neural networks. This connection exhibit a high is referred to the field of responsiveness of a node. Weighted sums from the weighted sums table are substituted to establish local connectivity. A type of neural network is convolutional neural networks. Layers of the convolutional neural network convolves the input.To make a feature map, the weight matrix used. In contrast to typical neural networks, the output feature map's values all have the same weights. This indicates that every node in the output detects the same pattern. CNNs lower the total amount of learnable parameters due to their local connection and shared weights. As a result, training is more

efficient. A convolutional neural network works by constructing a weight matrix in each layer that can extract the needed data from the input.

The weight, height, and number of channels are commonly considered three-dimensional inputs to a convolutional layer. To construct the feature output map, the first layer applies a set of 3D filters to all of the input channel to convolve the data in other words. Consider the following example: $x = (x_j)^{N-1}{}_{j=0}$, a one-dimensional input of size N with no zero padding. Convoluting each filters w1h for h = 1,..., M1 with the following input yields the output feature map of the first layer:

$$a^1(i, k) = (w^1{}_k * x)(i) = \Sigma\, (w^1{}_k\, (j)x(i - j)) \quad \text{where } j=-\infty \text{ to } \infty.$$

Features may be identified in a time invariant manner, but the number of trainable parameters is decreased, because all of the elements in feature map have the same weightage. The network has output after convolution, whose count and size depends on the size of filter used and final layer uses the number of filter.

To minimise errors in network output, the model's weights are trained depending on what our model needs to learn.

IV. Background

This paper performs comparative analysis on the performance of CNN, RNN, LSTM, BiLSTM and ARIMA in the time series forecasting of a financial dataset.

A. Data Set

The research uses CUDL dataset which has multiple stocks price data along with date. Using a single stock from the list of multiple stocks, dataset is divided into two different set known as training and testing set. Dataset of daily prices of 505 stocks that currently compose the S&P 500 and CASH value for the dates 01/01/2015 – 05/31/2018.

B. Training and Testing Data

The dataset has closing price of every company recorded. These closing price of the companies are fed into LSTM, ARIMA and BiLSTM models. The dataset was split into a training set and a test set, with 80% of the data going into the training set and 20% going into the test set.

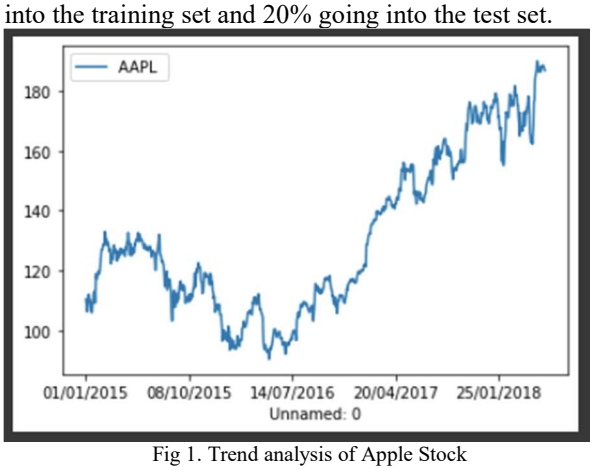


Fig 1. Trend analysis of Apple Stock

Figure 1 represents the trend of apple company stock from the year 2015 to 2018. The trend shows a strong uptrend from starting from the year 2016 to 2018.

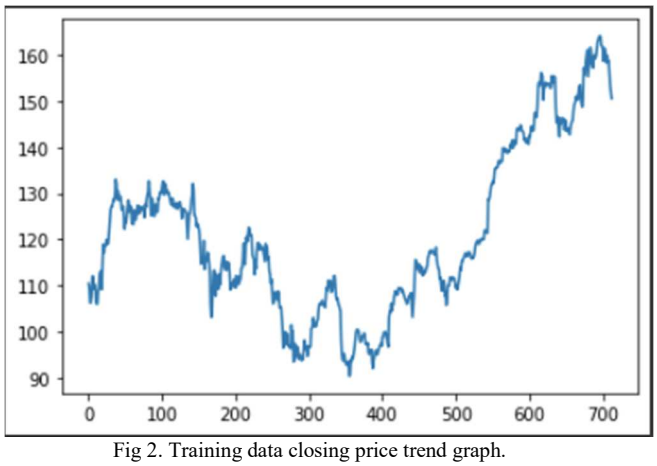


Fig 2. Training data closing price trend graph.

The figure 2 represents the training dataset for the apple stock closing price, training set has around 700 observations which is 80 percent of the dataset.

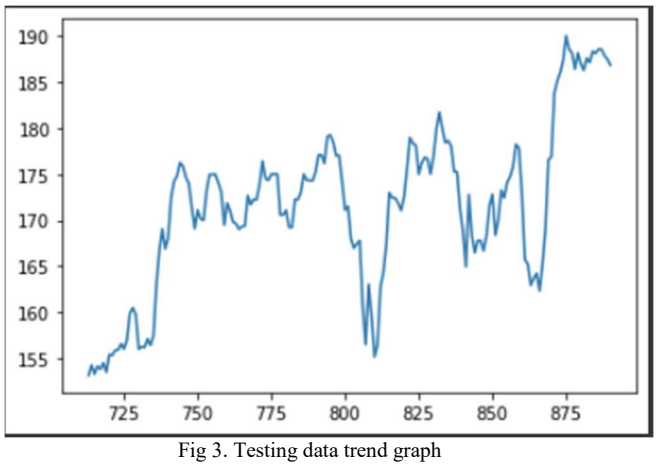


Fig 3. Testing data trend graph

Figure 3 represents the testing dataset for the apple stock closing price. The testing set is 20 percent of the full dataset which contains around 170 observations.

C. Model evaluation metrics

Deep learning algorithms typically reported the loss figures. Loss is a technical term denoting a penalty to be calculated for making a wrong prediction. In more detail, the loss value would be 0 if the model forecast is correct. To do this, the goal is to produce a set of biases and weights to bring the loss numbers down. Moreover, many researchers use the Root-Mean-Square-Error to evaluate how accurate their predictions are. RMSE is a statistic that assesses the distance between a company's actual and anticipated outcomes.

The RMSE formula is as follows:

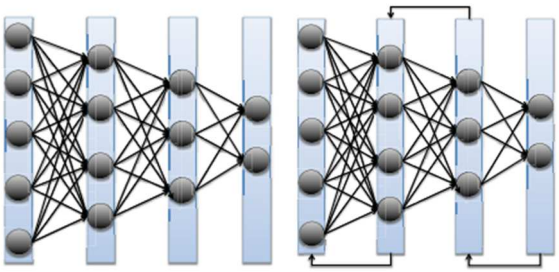
$$\mathrm{RMSD} = \sqrt{\frac{\sum_{i=1}^{N} (x_i - \hat{x}_i)^2}{N}} \tag{12}$$

X_i is real value and x_i is predicted value, where N is total number of observations. The key advantage of utilizing

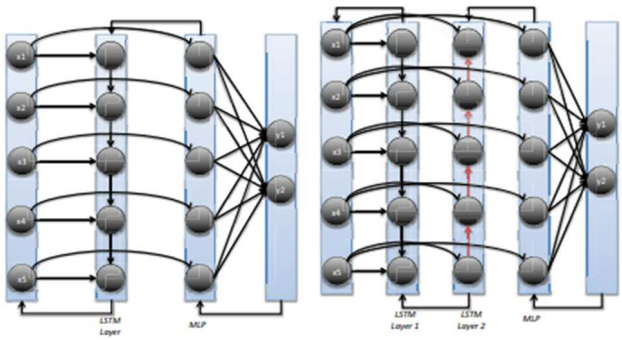
RMSE is that huge mistakes are penalized. The scores are also scaled in same measure as predicted values.

V. MODELS

The conventional feed-forward Artificial Neural Network. Figure 1a allow models to be trained by simply travelling in one direction without taking into consideration any prior input data feedback. ANN models, in particular, go completely new from input to output, ignoring the results of previously taught data. As a outcome, any layer's output has no bearing on the training process for that layer (i.e., no memory).



a) Feed Forward network *b) Feed backward*



c) LSTM *d) BiLSTM*

These neural networks have been proven to be effective at modelling the connection between input output points, and hence perform similarly to regression-based modelling. In different words, these network perform a functional connection mapping, where input data is translated to output. Convolutional Neural Network are examples of ANN model. Recurrent-based Neural Networks, on the other hand, recall portions of previous data in order to employ a feedback approach in which training proceeds from input to output but also from source to load. However, it also uses a network loop to save certain information and therefore acts as a memory (Figure 1(b)). Feedback based neural networks, in contrast to feedforward ANN networks and their position states updates over time until they find equilibrium and are therefore optimized. These states will stay in constant positon until fresh inputs arrive, which will alter the equilibrium. LSTM model as shown in Figure 1c is introduced as an extension to RNNs that allows them to retain lengthy input data, and therefore the link between lengthy input output data is characterized in terms of an extra dimension. To recall a long series of data, an LSTM network employs many gates, including an input gate, forget gate, and output gate. BiLSTM model [12] are variant of regular LSTMs in which the required model is trained both from input towards output and from output towards input. BiLSTM model initially feeds input data into one LSTM model, then repeats the training using one more LSTM model in the opposite order.

```
model = ARIMA(data, order=(5,1,5))

model_fit = model.fit()
# summary of fit model
print(model_fit.summary())

model = Sequential()
model.add(layers.SimpleRNN(neurons, batch_input_shape=(batchSize, X.shape[1], X.shape[2])))
model.add(layers.Dense(1))
model.compile(loss='mean_squared_error', optimizer='adam')

 #CNN model
 model = Sequential()
 model.add(layers.Conv1D(neurons,1, batch_input_shape=(batchSize, X.shape[1], X.shape[2])))
 model.add(layers.GlobalMaxPooling1D())
 model.add(layers.Dense(1))
 model.compile(loss='mean_squared_error', optimizer='adam')

model = Sequential()
model.add(LSTM(neurons, batch_input_shape=(batchSize, X.shape[1], X.shape[2]), stateful=True))
model.add(Dense(1))

model.compile(loss='mean_squared_error', optimizer='adam')

# Bi LSTM Model
model = Sequential()
model.add(Bidirectional(LSTM(neurons, batch_input_shape=(batchSize, X.shape[1], X.shape[2]), stateful=True)))
model.add(Dense(1))
model.compile(loss='mean_squared_error', optimizer='adam')
```

Fig 8. Models

The implementation of ARIMA, RNN, CNN, LSTM, BiLSTM model contain different first layer with the input shape. The loss used to evaluate the models are used is mean square error with Adam optimizer. Loss functions are used to determine how close an estimated value is to the genuine value. A loss function connects decisions to the costs they represent. Loss functions aren't fixed; they shift with the work at hand and the aim to be achieved. The RMSProp optimizer has been updated with Adam (short form for Adaptive Moment Estimation). Both the gradients and the second moments of the gradients are utilized as running averages in this optimization process.

VI. DISCUSSION

ARIMA model as compared to CNN and RNN models performed better and produced lower root mean square error. LSTM performed better than ARIMA in terms of performance and error metric, while BiLSTM out performs every model and gives the lowest error metric among other models.

BiLSTM's superior accuracy and performance over the normal unidirectional LSTM is understandable. Text parsing and next sequence predictions are the examples of it. However, it was uncertain if two times using numerical time series data and understanding from the outcomes would be sufficient. It would be more precise to anticipate the future and also the past sequence of events, because some circumstances may not exist, such as observable in the parsing of text.

A. Result and metrics

Root mean square is the metric which is suitable in the time series forecasting problem. Below is the result of the performance of the models measured using RMSE:

Algorithm	RMSE
ARIMA	1.028
CNN	1.317
RNN	1.558
LSTM	0.932
BiLSTM	0.773

TABLE I: RMSEs of algorithms

BiLSTM outperforms RNN, CNN, ARIMA and LSTM algorithm

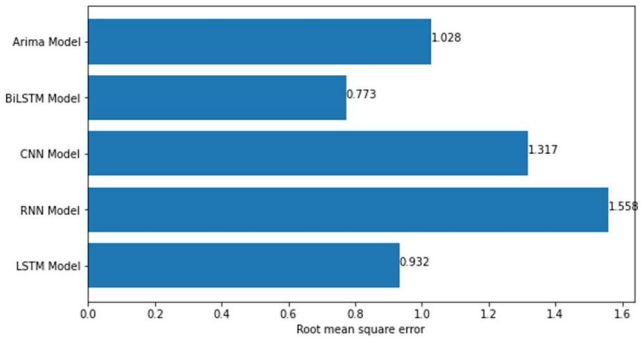


Fig 9. RMSE of algorithms

The above bar plot represents the root mean square error comparison of the ARIMA, BiLSTM, LSTM, CNN and RNN model.

The performance of LSTM, BiLSTM is better than other models. The Error can be further reduced on performing hyperparameter tuning on the model at initial state. Thus it is seen that BiLSTM outperforms other model in terms of RMSE for time series forecasting.

VII. CONCLUSION

With recent advances in the development of sophisticated machine learning-based approaches, these approaches, particularly deep learning algorithms, are getting importance among scientists. academics from a variety of disciplines. When compared to old procedures, the main question is that up to what level are the new algorithms accurate and effective. This research presented the findings of experiments in which the efficiency and accuracy of ARIMA, unidirectional LSTM, Convolution neural network , Recurrent Neural Network and bidirectional LSTM models, were studied and compared. BiLSTM is the variation of the LSTM which found to be better performing model. The further models can be improved by performing hyper parameter tuning and by addition of dense layers in the neural network.

As a result, for prediction problems in time series analysis, this research is suggests to utilize BiLSTM rather than LSTM. This study could be expanded to include multivariable and seasonal time series forecasting difficulties.

In accounting and business, the research provided in this research paper favors the adoption of the deep learning-based methods and technique. Deep learning may be used to address several other financial and economic prediction problems. The authors want to see how well deep learning works in a range of additional contexts and datasets with multiple amounts of features.

VIII. REFERENCES

[1] G. Box and G. Jenkins, "Time Series Analysis: Forecasting and Control," *San Francisco: Holden-Day,* 1970.

[2] A. M. Alonso and C. Garcia-Martos, " Time Series Analysis - Forecasting," *Universidad Carlos III de Madrid,,* 2012.

[3] M. Khashei and M. Bijari, " A Novel Hybridization of Artificial Neural Networks and ARIMA Models for Time Series forecasting," *Applied Soft Computing,* 2011.

[4] X. A. D. N. H. C. Krauss, "Deep neural networks, gradient boosted trees, random forests: Statistical arbitrage on the S&P 500," *FAU Discussion Papers in Economics,* 2016.

[5] S. J. S. J. Y. S. I. Lee, "A Deep Efficient Frontier Method for Optimal Investments," *Department of Computer Engineering Sejong University,* 2017.

[6] C. K. T. Fischera, "Deep Learning with Long Short-term Memory Networks for Financial Market Predictions," *FAU Discussion Papers in Economics,* 2017.

[7] I. E. Livieris and E. P. &. P. Pintelas, "A CNN–LSTM model for gold price time-series forecasting," *Emerging applications of Deep Learning and Spiking ANN,* 2020.

[8] S. Selvin, R. Vinayakumar, E. A. Gopalakrishnan, V. K. Menon and K. P. Soman, "Stock price prediction using LSTM, RNN and CNN-sliding window model," *IEEE,* 2017.

[9] N. T. a. A. S. N. S. S. Namini, "A Comparison of ARIMA and LSTM in Forecasting Time Series," *17th IEEE International Conference on Machine Learning and Applications (ICMLA),* 2018.

[10] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation,* pp. 1735-1780, 1997.

[11] P. Baldi, S. Brunak, P. Frasconi, G. Soda and G. Pollastr, "Exploiting the past and the future in protein secondary structure prediction," *Bioinformatics,* 1999.

[12] M. Schuster and K. K. Paliwa, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing,* 1997.