Stock Market Forecasting: Comparative analysis of SARIMA, CNN and LSTNet Models

Piyush Chaskar[1]

[1]Universiti Teknologi Malaysia, Malaysia (Affiliation1, Times New Roman, 10)

[2, 3]University of Waikato, New Zealand (Affiliation2, Times New Roman, 10)

*author1@utm.my (Affiliation email, Times New Roman, 10)

ABSTRACT

Indian stock market with market capitalization of $2.3 trillion, over 5500 listed firms on BSE and over 1700 listed firm on NSE has been a very promising market for investors all over the world. This market is tracked by two prominent indices Sensex and Nifty. Forecasting stock indices has been everyday struggle for financial analysts. The unpredictable nature of news and its impact on stock indices makes this task very difficult. Similarly, prediction of stock market crash, has been subject of study for decades. The aim of this work is to compare and examine accuracy of Long-and Short-term Time-arrangement Network (LSTM) with Seasonal ARIMA and Convolutional Neural Network (CNN) for forecasting returns of Indian stock market. For this study, historical data has been collected of Indian stock market from 2000 to 2020 and predictions are made for test data which is subset of collected data. Various models are compared using mean squared error and features of time series data. Stock return data has inherent characteristics such as its temporal nature, sequential nature, memory. Some of these characteristics are captured in LSTM and CNN which makes them effective for stock market forecasting. This study finds LSTM to yield least mean square error, thus making it most accurate amongst CNN and SARIMA in contrast to other published findings. This study provides comparison between statistical methods usually employed by finance analyst for analyzing stock returns. It highlights the fact the models used for individual company prediction may not be most accurate for index prediction. This study would be helpful in picking stocks for daily trade. Setting upper and lower limit for weekly trades.

Keywords

component, formatting, style, styling, insert

Article Received: 10 August 2020, Revised: 25 October 2020, Accepted: 18 November 2020

Literature Review

One of the most sought-after advice is financial world is the forecast of stock market returns. Many analysts, experts and gurus struggle every day to answer a simple question what the price of stock at a time in future would be. Several mathematicians, statisticians and economists have tried to propose theories which would predict stock movement to some degree of accuracy. With advent of computing various computational models were proposed to outperform the previous used linear and non-linear approaches. The difficulty lies in unpredictability of events which may impact the price of a stock.

Numerous techniques are employed by statisticians using historical data to predict future performance. Primarily autoregressive moving average (ARMA) and autoregressive integrated moving average (ARIMA) techniques have been fairly accurate. ARIMA characterizes univariate time series data. It employs unique Box-Jenkins stratagem for model selection. The flexibility of ARIMA model extends to accommodate various exponential smoothening. They can subsume auto-regression, moving average and ARMA. For higher dimensional time series data with multiple variables forecasting is difficult using ARIMA since it has high computational trade-off.

Mohapatra [1] in his paper highlighted various techniques such as autoregressive conditional heteroscedasticity (ARCH), generalized autoregressive conditional heteroscedasticity (GARCH), ARMA-EGARCH and Box and Jenkins. Techniques of ANN, SVM and fuzzy sets were classified as soft computing techniques by Mohapatra in his paper. The research paper also mentions other forecasting techniques Genetic algorithm and Ant Colony Optimization.

Mohapatra also compares forecasts using various techniques. He uses root mean square error (RSME) and mean absolute percentage error (MAPE) values as metric to determine which technique is optimum. Mohapatra research concludes that soft computing model FLANN (de-optimized) provides most accurate forecast daily, weekly, and monthly.

Artificial neural networks have been known to scientist for long. However only in late 1980s, they gained momentum and were recurrent in scientific and technical presentations. There after many research works has used these phenomenal techniques in the field of economics and finance. [2]

The first significant study of neural networks and its utility in stock market predictions was published by White [3]. Several decades of research has gone into perfecting and improving forecast using neural network since White's study. Neural network with a single layer is a generalization of linear function. The output is a direct result of linear function applied on input data. This basic neural network model system is additionally alluded to as the perceptron. However, in neural network with several layers, the neurons are orchestrated in layered design, in which the information and yield layers are isolated by a group of concealed layers. This system which is also known as feed-forward system is a multi-layered neural network.

An interesting and practical application of neural network yielded profits in simulations for the authors of [4].The authors developed a prediction system for Tokyo Stock Exchange Prices Indexes (TOPIX).This system used various relationships between technical and economic factors.

Several neural networks were utilized to learn these relationships and accurately forecast TOPIX.

To achieve similar feat as Asakawa, attempt was made by Erkam [5]. Data utilized was real exchange daily rate of NASDAQ. The metrics used to judge was mean square error (MSE) and mean absolute deviation (MAD). Erkam's research concluded that classical neural network model gives better estimator as compared to hybrid model.

Deep neural systems have had phenomenal success on expansive scope of issues. The Recurrent neural systems (RNN) models, for instance, have gotten generally well known for Natural language handling (NLP) research. Two variations of RNN specifically, LSTM [6] and the Gated Recurrent Unit (GRU) [7], have fundamentally improved the best in class execution in machine interpretation and discourse acknowledgment. They can successfully catch the implications of words dependent on the long haul and transient conditions among them in input reports. For recognizing pictures, CNN models have been very efficient. CNN models successfully extract features which are nearby and invariant, also known as shapelets. These shapelets can be extracted from different granular levels. Deep neural systems have gotten an expanding measure of consideration in examination of time series data. A generous part of the past work has been concentrating on time series characterization. For example, RNN models have been used for extricating patterns from medicinal sequential data and grouping the information with deference indicative classes. RNN has been applied to versatile information, for characterizing actions and activities. CNN models have been utilized in real life movement recognition by extracting features of move invariant neighborhood designs from input data as the highlights of grouping models. Deep neural systems have been successful for time series estimating i.e., the task of utilizing observed time series in the past to anticipate the obscure time series in the future – the bigger the time frame, the harder the issue.

For time series data with multiple variables a novel deep learning framework (LSTNet) was employed in [8].The paper observed that LSTNet captures both short term and long-term patterns in the data. It also fuses linear and non-linear models. LSTNet combines benefits of convolutional layer and repetitive layer to improve the accuracy of results. Lai provides empirical evidence to support the contention that LSTNet is indeed robust for predictions.

In research conducted by [9], it was concluded that artificial neural network (ANN) was efficient for forecasting stock market as compared to traditional models. ANN is a non-linear statistical tool [10]. Convolutional neural networks are intended to work with matrix structured input data, which have solid spatial conditions in nearby areas of the framework. Convolutional neural network will in general make comparative features from nearby locales with comparative patterns. A significant characteristic of convolutional neural network is a dot-product between a grid-structured set of weights and similar grid-structured input data. This data is captured from different localities of the spatial structure as input to the model. This sort of dot-product is helpful for analyzing information with a significant level of spatial or other localities. In this manner, convolutional neural systems are characterized as systems that have at least one convolutional operation in a layer, albeit most convolutional neural systems utilize this activity in many layers. In convolutional neural systems, the states in each layer are organized by a spatial matrix structure. These spatial connections are acquired starting with one layer then onto the next because each feature value depends on a small neighborhood spatial region in the previous layer. It is imperative to keep up these spatial connections among the matrix cells, on the grounds that the convolution activity and the change to the following layer is fundamentally reliant on these connections.

In their paper [11], determined the types of deep neural architectures to be used based on type of applications. Multi-layer perceptron, Recursive Neural Networks, LSTM, CNN were under the purview of their study. Stock market is characterized by uncertain and unforeseeable events and reactions. Therefore, forecasting inherently takes higher risk compared to other areas. The added risk factor makes the application of deep-neural models in finance even more lucrative [7]. Deep learning algorithms have proved to be more accurate than conventional learning algorithms for stock market forecasting. Recurrent deep neural network model was tested on ten Nikkei companies return. The model was used to test trend of return in by [12].

In this work [13], four deep learning models, MLP, RNN, LSTM and CNN were used. They modelled these architectures for a company listed on National Stock Exchange of India (NSE). That model was then used to predict stock prices for 5 other companies on New York Stock Exchange (NYSE) and NSE India. Their research concluded that there are some intrinsic common dynamics which enables model trained on NSE to predict accurately for companies listed on NYSE. They also concluded that CNN outperformed the other models. In comparison between ARIMA and neural networks, ARIMA was clearly inefficient.

Advancing on the previous work [14], tired a model independent approach. The data was not fitted into a model. [14] attempted to identify inherent dynamics existing in the data using deep learning models. They tried predicting stock prices of companies listed on NSE by using a sliding window approach. Percentage error was the metric utilized to adjudicated amongst RNN, LSTM and CNN Models. Research concluded that CNN is best model amongst the other two to capture hidden dynamics of stock movement for various companies.

In this paper we compare three models namely, Long-and Short-term Time-arrangement Network (LSTM), Seasonal ARIMA and Convolutional Neural Network (CNN) Models. We try to predict performance of these model on National Stock Exchange: Index Nifty 500.

Research Methodology

A. Research Data

For this research, data is collected from CEIC. Data consists of all possible economic measures of Indian stock market. Total of 353 variables were collected over the period of 20 years. These are all possible indicators of stock market returns over last 20 years. The duration for the purpose of this study is considered from January 1st, 2000 to February 28th, 2020.

B. Data Cleaning Specifications

The Indian stock market does not trade on weekend. There are however exceptional weekends where Indian stock market is open. For the purpose, of this analysis and consistency of data, weekends have been dropped from the data. As a rule, all columns with more than 10% missing values cannot be considered as they do not contain complete information. Therefore, such variables are also dropped. Linear interpolation is used to estimate missing values for variables with 10 % or less missing values.

C. Data Cleaning Specifications

For this study we have chosen to predict returns of National Stock Exchange: Nifty 500. This variable therefore is considered as dependent variable. All other variables are independent variables, which may or may not impact the dependent variables.

D. Methodology

After cleaning the data, we test it for stationarity. Stationarity is a property of time series data which suggest that distributional properties of data are constant across time. Data must be stationary for forecasting. Non-stationary data suggest the past data is not indicative of future returns. Stationarity is tested for National Stock Exchange: Nifty 500 variable, as it is the one which we are trying to predict.
To test whether data is stationary, p-value is evaluated using Dickey-Fuller Test. P-value calculated from Dickey-Fuller Test was found to be 0.964552. Furthermore, the plot rolling mean and rolling standard deviation of the dependent variable displays upward trend. Therefore, it can be concluded that data is non-stationary.

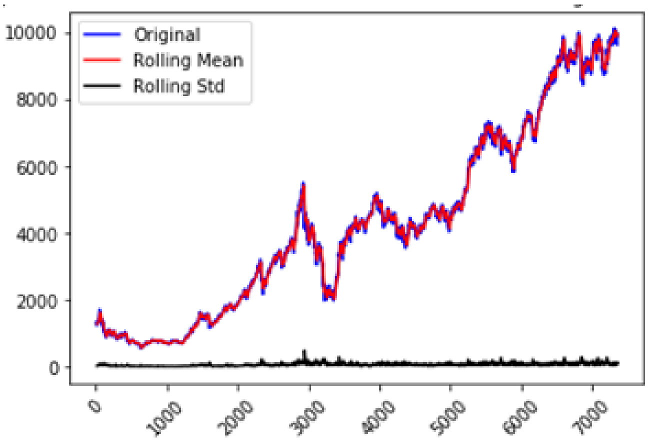

Fig. 1: **Rolling Mean & Standard Deviation for NSE Index Nifty 500**

To make data stationary easiest way is to calculate the first difference. For Indices, data is made stationary by finding the difference in the percentage change vis-a-vis previous time- period. For percentages, similarly subtraction from previous time-period is calculated. Again Dick-Fuller Test is conducted, p-value of 4.842084e-30 is lower than 0.05 therefore data is stationary. Plot of rolling mean average and rolling standard deviation also displays straight line which indicate stationary data.

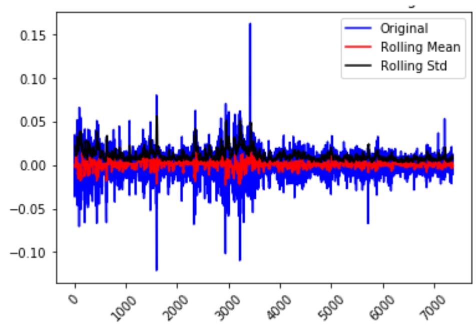

Fig. 2: Rolling Mean & Std. Devi. for NSE Index Nifty 500 adjusted

Time series data is auto regressive in nature, which means that current values are dependent of previous values. Therefore, lagged versions of each independent variable are created to deal with autoregressive nature.
The maximum number of lags of the dependent variable to use can be decided with help of Autocorrelation Function (ACF) and/or Partial Autocorrelation Function (PACF) plot or decided heuristically based on domain specific cycles (ex: business cycle, seasons, etc.). Minimum number of lags depends on forecasting horizon. If forecast is for h steps ahead, first h lags are excluded. Below plots of ACF and PACF suggest high correlation for first lag if prediction are made for next day of Nifty 500 Index.

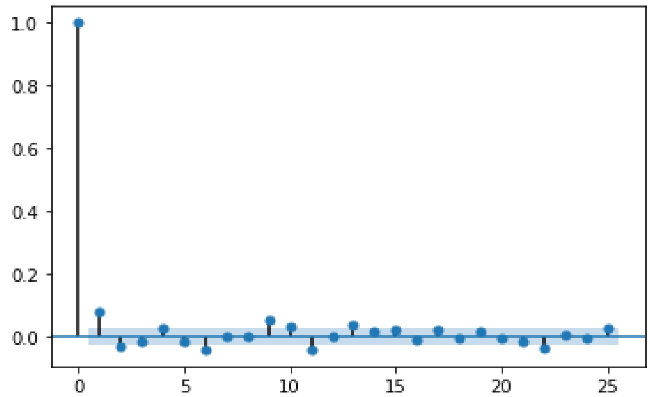

Fig. 3: Observed Partial Autocorrelation

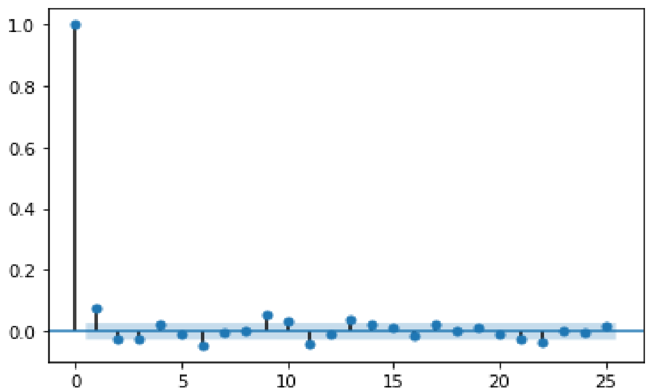

Fig. 4: Observed Autocorrelation

For model implementation, lagged versions of independent variables are taken into consideration, excluding date.

The objective of any machine learning algorithm is to predict the dependent variable accurately. Minimizing difference between actual value and predicted value would help in identifying best method to predict stock market returns. Accuracy of a model is evaluated on out of sample data points, as overfitting a model will result in arbitrarily high accuracy. Therefore, data must be split into training and test data. Testing data is further split into validation data and test data. This is primarily done to avoid data leakage. Data leakage is a phenomenon which describes overfitting of model on out-of-sample test data. Overfitting results in poor generalizability therefore validation data is used for model selection. This selected model is then used to calculate MSE on testing data.

There are many ways in which data can be organized. For instance cross validation using bootstrapping, rolling window in which a window of fixed size is move ahead by successive observation, fixed window wherein training and testing samples are divided on basis of dates or expanding window where window size itself is not constant. For the purpose, of this study, a fixed window approach is used. However, for time series analysis rolling or expanding window approach is highly recommended.

Model of machine learning require normalized data. The learning methods are not scale invariant. Normalization is also required to prevent data leakage. We scale the data between negative one and positive one to be perfect input for a Convolutional Neural Network (CNN) and Long Short-Term Memory Network (LSTM). Data is scaled for all methods considered in this study so that errors are comparable between the methods. Mean and standard deviation of normalized data and rescaling parameters are used to normalize and rescale testing and validation data. This ensure that the out-of-sample data does not have data leakage or forward bias.

Methodology for the study is inspired from [14]. Three models architectures SARIMA, CNN and LSTM are used for the purpose of this work. Recurrent neural network enables connections to form a directed round amongst the computational units. It uses internal memory to understand the trends/sequence of input data. Each such computing unit is assigned a real valued activation which is time varying and weight which can be changed.

Equation 1 is used by RNNs to define value of weights or other hidden units recursively over a graph-like structure.

$$h^t = f\{h^{t-1}, x^t, \theta\} \qquad (1)$$

A modified and advanced version of RNN was introduced by (Schmidhuber, 1997). Difference between the two lies at the hidden layer where LSTM cells are introduced. These cells are composed of gates. These gates act like values for flow of inputs. Four types of gates are used by LSTM, input, cell state, forget and output. LSTM additionally has sigmoid function, tanh function and dot product operations. Each gate performs specific operations such as input gate stores the inputs, cell state has ability to discard or add information, forget get usually determines the proportion of information to be allowed and finally output gate consists of predicted values by LSTM models. Primary function of sigmoid operation is to generate number between 0 and 1. Thus describing how much each component should be allowed inside. Tanh's primary function is to generate additional vectors. Mathematically equation 2-6 describe the function of gates in LSTM Model.

$$f_t = \sigma\{w_f.[h_{t-1}, x_t] + b_f\} \qquad (2)$$
$$i_t = \sigma\{w_i.[h_{t-1}, x_t] + b_i\} \qquad (3)$$
$$c_t = \tanh\{w_c.[h_{t-1}, x_t] + b_c\} \qquad (4)$$
$$o_t = \sigma\{w_o.[h_{t-1}, x_t] + b_o\} \qquad (5)$$
$$h_t = o_t * \tanh c_t \qquad (6)$$

TABLE I: VARIABLE DESCRITPION

Variable	**Descriptions**
x_t	Input
h_t	Output
c_t	Cell state
f_t	Forget gate
i_t	Input gate
o_t	Output gate
W,b	Parameter matrix and vector

CNN works like a grid, it maps data spatially. Time series data is treated as one dimensional whereas image is treated two-dimensional data. A specialized mathematical operation called as convolution which is a linear operation is implement in CNN. CNN architectures mainly focus on the given input sequence and does use memory of past inputs during learning process.

For testing accuracy of various methods, we employ mean square error (MSE). Mean squared Error (MSE) is commonly used to test accuracy of time series models. Higher MSE represent bad performance and lower MSE represents good performance. For comparing models in this study, MSE metric is used for gauging the accuracy.

Results and Analysis

The mean squared error is most popular metrics for evaluating accuracy of prediction models. Mean square error is calculated by first finding difference between estimated quantities and actual quantities. This difference is then squared. Finally, we take mean of all squared differences. Risk function arises due to uncertainty. It is an estimator of function of loss as function of a rule which makes decision. MSE is one such risk function. MSE is an estimator of square of error loss. MSE is often positive because estimates do not account for randomness. In this research MSE is calculated for 3 Models namely SARIMA, CNN and LSTM. MSE for validation data and MSE for testing data are both indicators of accuracy of the model. However, MSE for test data is given preference over MSE for validation data.

TABLE I I: MSE RESULTS

MSE	Seasonal ARIMA	CNN	LSTM
Validation	0.0028029330	0.0021614244	0.0022043467
Testing	0.0038464071	0.0033692841	0.0033005664

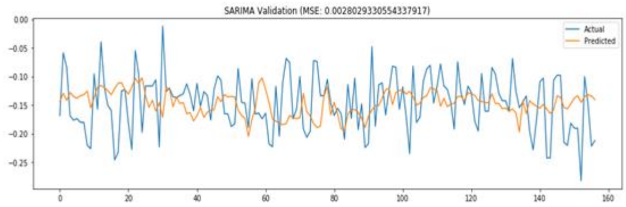

Fig. 5: NSE Nifty 500 return forecast using SARIMA on validation data

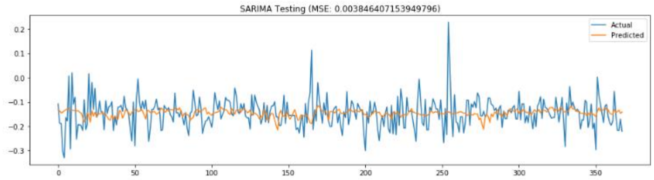

Fig. 6: NSE Nifty 500 return forecast using SARIMA on testing data

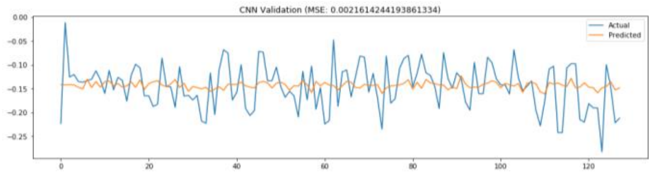

Fig. 7: NSE Nifty 500 return forecast using CNN on validation data

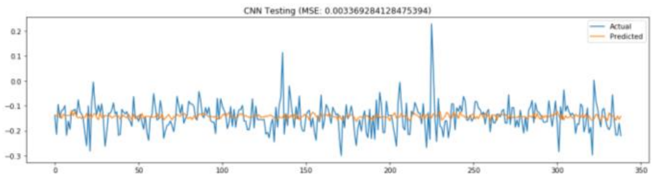

Fig. 8: NSE Nifty 500 return forecast using CNN on testing data

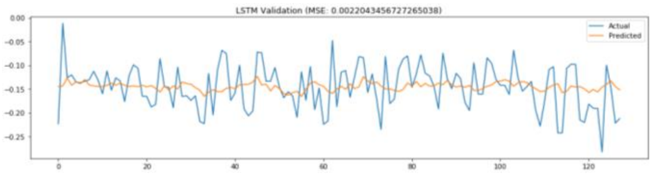

Fig. 9: NSE Nifty 500 return forecast using LSTM on validation data

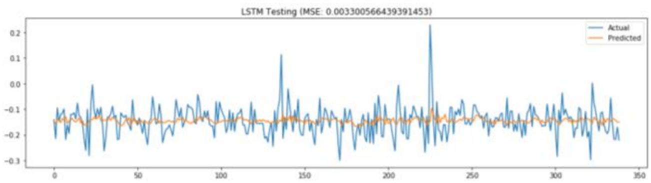

Fig. 10: NSE Nifty 500 return forecast using LSTM on testing data

Discussion

The In previous published work [14] and [10] for prediction of stock prices of a company listed on NSE it was found that CNN models yielded more accurate predictions as compared to predictions yielded by LSTM. However, results obtained using this work contrasts these previous studies.

To start with ordinary least square (OLS) method was given a consideration for predicting stock market. However certain shortcomings of the method proved to be detrimental in rejecting the method.

To overcome shortcoming of OLS, ARIMA models were considered. ARIMA which is autoregressive integrated moving average, describe how an observation is related to antecedent observation. Seasonal ARIMA is capable of modelling data which has seasonality component in it. Seasonal variations in single variable data, autoregressive moving averages and various trends/pattern are easily modelled by SARIMA.

As discussed early, multiple studies have been conducted proving the utility of neural networks in time series forecasting. Neural network models are universal function approximators. These neural networks typically have 3 layers, namely – input layer, computational layer, and output layer. Each layer consists of nodes and consequent layers are connected through weights. Linear regression as a neural network treats the data cross sectionally. It does not find non-linearities in the data thus limiting usage. It cannot gauge information from the spatial and sequential nature of time series data. A Deep neural network creates non-linearities. It has multiple hidden computational layers and can infer information from time series data. In this study we first forecast returns using one dimensional convolutional neural network (1D-CNNs) and then forecast returns using Long short-term memory networks (LSTM)

While OLS and SARIMA are dependent on manually selected independent variables, CNNs and LSTM can create engineered variables. OLS and SARIMA do not create new variables using existing independent variables. CNNs and LSTMs can learn an internal representation of the time series data. They achieve similar performance to fit on a version of the dataset with engineered variables. In deep neural network input layer preserves the temporal structure of time series data, which means initial weights of a neural network are assigned randomly. The hidden layer which consists on complex computational layer uses two types of network processing the data differently.

Critical information is stored in temporal structure of the data. This information is captured by convolutional neural network (CNN) by using filters. The patterns of spatially nearby data is learned by CNNs filter at the input layer. For instance, one filter could find peaks, troughs, or linear trends. Feature map is the information extracted by the filters. Each additional feature results in a new feature map extracting a more complex feature. A filter moves along one dimension at a time in time series data. A sub sampling layer which reduces the noise in the learned features is also present in CNN layer. The sub sampling layer is followed by a regression layer.

To build a neural network that can remember and extract information from a sequence Long short-term memory (LSTM) network is used. LSTM remembers sequential data. It draws information from such sequential time series data. It can remember long series of data thus deriving information from such sequences. LSTMs belong to Recurrent neural network family. RNNs is a larger class of neural network which has memory. While LSTMs may learn from longer networks, all RNNs may not. Data is revealed to the hidden nodes in a sequential fashion to build short and long memories. This feature is used in Long- and short-term memory models. LSTM cells are part of each LSTM node. The output of each 'gate' provides for a condition upon which each LSTM cell is exposed to subsequent element in the sequence. Thus, updating long term and short-term memory. A sigmoid/logistic activation function is term as neural network gate. This function determines how the sequential memory is updated by inputs. Inputs are the current elements of a sequence and the output elements of previous long-term memory or hidden memory. These element in the hidden memory are hidden from previous member of the sequence. Output 0 from a gate implies no information is transferred. Similarly output 1 from a gate implies all the information is transferred. LSTM uses multiple gates to enhance its memory. These gates are namely, The Forget Gate /or Remember Vector, Candidate Gate, Input Gate or Save Vector and Output gate or Focus Vector. The output of an LSTM node is update to the working memory. This updated working memory is the new representation of the time series data which LSTM model has learnt. Working memory acts as new feature as input in the regression layer.

LSTM approaches the problem by capturing two important aspects of data temporal nature and relation with long haul conditions of indices. It employs convolutional layer to deal with first aspect and repetitive layer to deal with the second. Repetitive structure remembers long-haul relations between data. Thus, making the advancement simpler as it uses property of time series signals. Two models i.e. traditional autoregressive linear model and indirect neural network model are fused together by LSTM. This fusion makes non-linear deep learning model increasingly vigorous for time series data which is non-stationary in nature. In the investigation for the purpose of the study, it was found that LSTM reliably beats CNN and SARIMA.

Primary difference between this study and previous work i.e. [14] and [10] is that this study focuses on predicting prices of Index rather than individual company. [14] employs moving window approach, however for the purpose of this study fixed window approach is considered. Time frame taken into consideration for both previous works in shorter as compared to 20 years of time frame considered in this study. It is evident by comparison the short-term volatility may be best captured by CNN, which does not use memory while learning, however long-term returns are best predicted by LSTM which has the memory feature.

Conclusions and recommendations

For accurate forecasting, certain features of time series data and model are considered. Features such as creation of calculated, engineered featured are enable the model to compute faster and retain critical information. Temporal structure of data refers to timely nature of data. In time series analysis temporal structure is vital for any forecast.

It is possible to have a smaller number of observations and a greater number of independent variables in time series analysis. As in the case of stock market returns, model capable of handling such data is useful. Certain models we have considered in our analysis are useful only for predicting data which is linear in nature. Stock market returns, in some cases have been exponential. Non-linear fall during stock crash or market crash are also common occurrences. Therefore, models like LSTM and CNNs can provide better degree of accuracy. Certain models require data to be stable, such requirement puts constraints on input data. To make data stationary, information is lost. LSTM and SARIMA are only two models in this analysis which consider data sequentially for analysis. Only these two-model account for relationships which may be found due to sequential nature of the data.

TABLE III: TIME SERIES FEATURES FOR EACH MODEL

Nature of Time Series Data	Ordinary least squares	Seasonal ARIMA	CNN	LSTM
Creation of new features	No	No	Feature Map	Working Memory
Data's temporal structure	No	No	Yes	Yes
Number of Observations < Number of Independent variables	No	No	Yes	Yes
Linear in nature	Yes	Yes	Non-Linear	Non-Linear
Stationary Requirement	Yes	Non-Stationary	Non-Stationary	Non-Stationary
Display Trends	No	Only for dependent variable	Yes	Yes
Sequential nature of data	No	Only for dependent variable	No	Yes

It is apparent that LSTM models take maximum features into consideration for stock returns forecasting. LSTM Models are better suited forecasting technique for stock market returns while taking into consideration feature of the data and least MSE.

Limitations

Each Model has limitations and benefits. There is always tradeoff between speed of computation and accuracy of the model. Limitations of OLS, SARIMA, CNN, LSTM are highlighted below.

Ordinary least square (OLS) cannot deal with sequentially ordered data since order does not matter for OLS. It does not consider correlation between independent variables. It does not deal with spurious correlations also. Data which is not stationary gives unreliable results in OLS Model.

SARIMA models on the other hands have potential for error propagation. They tend to give poor results for turning points such as market corrections or market crash. These models also have tendency to move towards mean. As we forecast further into the future, the forecast converges to the mean of the data. Thus, a better model is required for forecasting.

Shortcoming of CNNs for time series analysis is that they miss out information from the sequential nature of

independent data. Sequential nature provides key information for forecasting as yesterday's stock returns would be decent indicator of today's stock return under same environmental conditions. However, CNNs do not consider this information for forecasting.
LSTMs models are slower to train. Thus, a tradeoff is made in training model which takes more time and computational memory. Accuracy of LSTM models is high, which may also result in overfitting the model. LSTMs Models are unable to implement dropouts. Initialization weights in LSTM Models play vital role, as the model is sensitive for random weight during initialization and can yield varied results each time.

References

[1] P. &. R. A. &. P. T. K. Mohapatra, "Indian Stock Market Prediction Using Differential Evolutionary Neural Network Model," International Journal of Electronics Communication and Computer Technology (IJECCT) , pp. Volume 2, pg 159-166, 2012.

[2] A. Hamid, "Primer On using Neural Networks for Forecasting Market Variables," in Proc. of the Conference at School of Business, Southern New Hamshire University, 2004.

[3] White, "Economic prediction using Neural Networks: The case of IBM daily stock," in In Proc. of the IEEE International Conference on Neural Networks, 1988.

[4] T. K. a. K. Asakawa, "Stock Market Prediction System with Modular Neural Networks," in 1990 IJCNN International Joint Conference on Neural Networks, San Diego, CA, USA,, 17-21 June 1990.

[5] G. K. T. U. Erkam Guresena, "Using artificial neural network models in stock market index prediction," Expert Systems with Applications, pp. Volume 38, Issue 8, Pages 10389-10397, August 2011.

[6] S. Hochreiter, "LONG SHORT-TERM MEMORY," Neural Computation, pp. 1735-1780, 1997.

[7] P. N. G. a. W. J. H. Heaton J. B., "Deep learning for finance: deep portfolios," Applied Stochastic Models in Business and Industry 33 (1), pp. 3-12, 2017.

[8] W.-C. C. Y. Y. H. L. Guokun Lai, "Modeling Long- and Short-Term Temporal Patterns with Deep Neural Networks," SIGIR, arXiv:1703.07015 [cs.LG], 2018.

[9] D. R. a. P. D.D, "Application of Artificial Neural Network for stock market predictions," International Journal of Machine Intelligence, ISSN: 0975–2927, pp. Volume 2, Issue 2, pp-14-17, 2010.

[10] A. A. a. S. V. N. Rather A. M., "Recurrent neural network and a hybrid model for prediction of stock returns.," Expert Systems with Applications 42 (6), pp. 3234-3241, 2015.

[11] I. J. G. a. A. C. Y. Bengio, "Deep learning," Nature vol. 521, pp. 436-444, 2015.

[12] F. K. S. K. U. K. Yoshihara A., "Predicting Stock Market Trends by Recurrent Deep Neural Networks," in Pacific Rim International Conference on Artificial Intelligence, 2014.

[13] E. A. G. V. K. M. K. P. S. M Hiransha, "NSE Stock Market Prediction Using Deep-Learning Models," Procedia Computer Science 132, ELseveir, pp. 1351-1362, 2018.

[14] R. V. E. A. G. V. K. M. a. K. P. S. S. Selvin, "Stock price prediction using LSTM, RNN and CNN-sliding window model," in 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, 2017.

[15] W.-X. a. S. D. Zhou, "2000-2003 Real Estate Bubble in the UK," Physica A: Statistical Mechanics and its Applications, pp. 249-263, 2003.

[16] W.-X. S. D. Zhou, "Nonparametric analyses of log-periodic precursors to financial crashes," International Journal of Modern Physics C, p. 1107–1125, 2003.

[17] W.-X. Z. ,. D. S. R. W. Zhi-Qiang Jiang, "Bubble diagnosis and prediction of the 2005–2007 and 2008–2009," Journal of Economic Behavior & Organization, p. 149–162, 2010.

[18] S. D. Z. W.-X. Woodard R, "The 2006-2008 oil bubble: evidence of speculation and prediction," Physica A 388, pp. 1571-1576, 2009.

[19] H. W. K. a. A. W. Tucker, "Nonlinear programming," in 2th Berkeley Symp. Mathematical Statistics and Probabilistics, Berkeley, CA, 1951.

[20] L. J. C. a. F. E. H. Tay, "Support Vector Machine With Adaptive Parameters in Financial Time Series Forecasting," IEEE TRANSACTIONS ON NEURAL NETWORKS, VOL. 14, NO. 6, pp. 1503-1518, 2003.

[21] D. Sornette, Why Stock Markets Crash: Critical Events in Complex Financial Systems., Princeton: Princeton University Press, 2003.

[22] S. H. a. J. Schmidhuber, "Long short-term memory," Neural Computation, Vol 9, p. 1735–1780, 1997.

[23] G. R.S., "Econometric tests of asset price bubbles: taking stock," Journal of Economic Surveys 22 (1), p. 166–186, 2008.

[24] J. L.Elman, "Finding structure in time," Cognitive Science Volume 14, Issue 2, pp. 179-211, 1990.

[25] C. Kindleberger, Manias, Panics, and Crashes: A History of Financial Crises, 4th ed., New York: Wiley, 2000.

[26] C. G. K. C. Y. B. Junyoung Chung, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," arXiv:1412.3555 [cs.NE], 2014.

[27] A. Greenspan, "Economic volatility," 30 August 2002. [Online]. Available: https://www.federalreserve.gov/boarddocs/speeches/2002/20020830/default.htm.

[28] G.Rotundo, "Neural networks for large financial crashes forecast," Physica A: Statistical Mechanics and its Applications, pp. Volume 344, Issues 1–2, Pages 77-80, December 2004.

[29] F. E. H. T. a. L. J. Cao, "Application of support vector machines in financial time series forecasting," Omega, vol. 29, no. 4, p. 309–317, 2001.