State-of-the-art vs prominent models: An empirical analysis of various neural networks on stock market prediction

Mansi Dhawan[a,1] , Tripti Aggrawal[a,1,*]

a Independent Scholar

1 Equal contribution: co-authors listed in alphabetical order

Abstract

Stock trading has always been a crucial and risky way of making money, requiring a profound understanding of the market and the data. Hence, stock market price prediction has always been a topic of interest for the research community. Existing literature has used legions of ways to accurately predict the price of stocks using sentimental analysis and fundamental and technical indicators, combined with the multitudinous linear, machine, and deep learning models. The existing research primarily focuses on classic univariate linear models like ARIMA, machine learning models including regression analysis and classification strategies, and traditional deep learning methods like LSTM; these have been the celebrity models in the stock price prediction problem. In the past few years, Recurrent Neural Networks (RNN) models have become synonymous with time series forecasting. Moreover, numerous new state-of-the-art models have also been gaining focus for time-series forecasting. This paper compares these new state-of-the-art models, including Temporal Fusion Transformer (TFT), N-BEATS, and Temporal Convolution Network, with prominent models, LSTM and GRU. Two years' worth of historical data from different securities listed on the National Stock Exchange(NSE) of India is fed into these models to predict near-future closing prices. The comparison of the models is made using four performance metrics: Root Mean Square Error, Mean Absolute Error, Mean Absolute Percentage Error, and R Squared Score. The results showed that LSTM and GRU outperformed every other model with the slightest error. Moreover, TFT outperformed the state-of-the-art models and had somewhat comparable performance with LSTM and GRU, but not better!

Keywords: [Deep Learning, Time series Forecasting, Temporal Fusion Transformer, LSTM, GRU, Stock Market]

1. Introduction

The stock market is used for trading shares of public listed companies. Firstly, in the primary market, companies float their claim to the public via an initial public offer (IPO). Afterward, the regulated market called the secondary market is used to deal in these securities or stocks. In the secondary market, one investor buys shares from another at the prevailing market price or the agreed price by the buyer and seller.

Stock markets tend to be unpredictable, and stock prices are essentially discrete-time series models based on well-defined numbers collected at successive points at regular intervals.

Due to these characteristics, financial data necessarily possess a rather convulsive structure which often makes it hard to find reliable patterns.

* Corresponding author Email Address: tripti.aggrawal@outlook.com

Nti IK et al. (2020) did a critical review on 122 research works from 2007–2018 and identified that these mainly used technical (66%), fundamental (23%), and combined analyses (11%). Also, Support vector machine and artificial neural network were the most popular algorithms for stock market prediction, moreover, Simple Moving Average(SMA), Exponential Moving Average (EMA), Moving Average Convergence Divergence (MACD), Relative Strength Index (RSI), and rate of change (ROC) were the most common technical indicators used.

Classic Time Series models are simple, and generally univariate, and require time-consuming pre- and post-processing steps to get accurate results. These models learn from past observations and predict future values using only recent history, for instance, Autoregression (AR), Moving Average (MA), Autoregressive Integrated Moving Average (ARIMA), and Simple Exponential Smoothing (SES). The same approaches are picked by Ariyo AA et al.(2014), Liu M et al. (2020). Machine learning models- regression and classification models, have also been used in time series forecasting. Some models include Support Vector Regression (SVR), Naive Bayes, XG Boost, Artificial Neural Network (ANN), etc. proposed in Sharma et al. (2017).

Modeling such convulsive systems requires deep learning algorithms capable of finding overridden structures and relations within the data and predicting their effect on future data. It can deal with complex structures efficiently and extract relationships that further increase the accuracy of the generated results. Deep learning models not only overcome the issues encountered in classical and machine learning models but better handle the complexity and nuances of time series forecasting and, thus, obtain significantly improved results. Greco M et al. (2021) illustrated predicted the closing prices of Exprivia S.p.A. in the Stock Market. and reported that deep learning architecture outperformed the traditional methods and multi-variate setting can significantly outperform the univariate one.

This paper compares five deep learning models on ten stocks listed on National Stock Exchange.

Some key points regarding the same are as follows:

- Two stocks were picked from five different sectors from NSE. This was to achieve diversity in our test portfolio.
- We also factored in technical indicators and each stock's closing price. We picked an optimized set of indicators that removed redundancy and rendered maximum insight or patterns in terms of volatility, trend, momentum, and volume.
- We only used two years' worth of historical data as opposed to bulky historical data used in most of the research, as the stock market is quite dynamic and trends change frequently. Moreover, past data doesn't have much influence or significance for short-term prediction or near-future prediction.
- We compared three new state-of-the-art models with two prominent deep learning models and have documented the results.

The rest of the paper is organized as follows: Section 2 briefly reviews the existing literature for stock market price prediction. Section 3 briefly explains the applied methodologies, models, and corresponding metrics chosen for evaluation. Section 4 discusses the dataset, a crisp step-by-step procedure, and the experimental results. Section 5 presents some conclusions and future directions.

2. Related Work

Deep learning models have proved to be better than classical linear models like ARIMA which are univariate and cannot capture the underlying dynamics and features of a time series data in M H et al. (2018) proposed that neural networks model like Convolutional Neural Networks (CNN), LSTM, and RNN outperformed classic models like ARIMA in predicting prices for securities listed on NSE and New York Stock Exchange (NYSE).

Moreover, deep learning models have proven more efficient and accurate than machine learning models as well. In Nabipour M et al. (2020) prices are predicted of securities/stock from four stock market groups from the Tehran stock exchange using nine machine learning models -Decision Tree, Random Forest, Adaptive Boosting (Adaboost), eXtreme Gradient Boosting (XGBoost), Support Vector Classifier (SVC), Naïve Bayes, K-Nearest Neighbors (KNN), Logistic Regression, ANN and two deep learning methods RNN and Long short-term memory (LSTM). They used ten technical indicators in continuous and discrete forms on ten years of historical data as the input values. The final results show that deep learning methods LSTM and RNN outperformed all the machine learning models in both continuous and binary data.

Technical indicators also have a significant influence on the quality of stock price prediction. Oriani FB et al. (2016) evaluated the impact of technical indicators by comparing 12 technical indicators, including RSI, Weighted Moving Average (WMA), and Average Directional Index (ADX), for one-day-ahead closing prices prediction using Multilayer Perceptrons (MLPs). The results show that mainly lagging indicators might lead to high-quality forecasts of stock closing prices when combined with the close price. It is also crucial to pick the right combination of indicators to get an accurate buy/sell signal and avoid redundancy and false signals. The key to achieving this is to reduce multicollinearity. As described in Franke et al. (2010), perfect collinearity results when two or more predictors are entirely redundant. Correlations above 0.8 or 0.9 between predictor variables are often interpreted as excessively collinear. Hence, highly correlated indicators should be avoided.

LSTM was first proposed in Hochreiter et al. (1997) and has been applied in various areas, including time-series prediction, music composition, and speech processing. Althelaya et al. (2018) conducted experiments to evaluate the performance of deep RNN architectures using stacked LSTM, bidirectional LSTM, bidirectional GRU, and stacked Gated Recurrent Unit (GRU) for stock market forecasting. The results showed that S stacked LSTM produced the highest short- and long-term forecasting performance.

The authors of Samarawickrama AJP et al. (2017) focused on using models to predict daily stock prices of stocks of Colombo Stock Exchange (CSE) based on the Recurrent Neural Network (RNN) Simple Recurrent Neural Network (SRNN), GRU and LSTM architectures were employed in building models. SRNN and LSTM networks generally produce lower errors compared with feedforward networks.

TFT, an attention-based Deep Neural Network optimized for outstanding performance and interpretability, was first proposed in Lim B et al.(2021). Since then, it has been gaining attention in different problem statements involving time series forecasting. Authors of Wu B et al. (2020) used TFT to predict wind speed forecasting. Similarly, Zhang H et al. (2022) predicted freeway speed prediction in intelligent transportation management using TFT. Hu X et al. (2021) is one of the first pieces of literature to illustrate the use of state-of-the-art Temporal Fusion Transformers for stock market prediction. He compared it with two popular deep learning and machine learning model, LSTM and Support Vector Regression respectively. Results showed that TFT achieved the lowest errors and outperformed the other two.

Liu Y et al. (2019) applied the temporal convolutional network to various time series prediction problems. They use the model for stock closing price prediction, Mackey-Glass time-series data prediction, PM2.5 prediction, and appliances energy prediction. The results documented reveal that compared with the traditional methods, LSTM, and GRU, the temporal convolutional network, gated temporal convolutional network, and multi-channel gated temporal convolution network converge faster and had better performance. TCN is significantly better than LSTM and GRU in single-factor time series prediction, but the advantage is not apparent in multi-factor time series prediction, even worse than GRU.

Sbrana A et al. (2020) proposed an extension of N-BEATS by adding a Recurrent Neural Network (RNN) in the proposed paper interpret its residual outputs, named N-BEATS-RNN. They performed it on the M4 competition dataset. Results show that in most cases, original N-BEATS outperformed N-BEATS-RNN.

3. Materials and Methodology

3.1 Technical Indicators and its types

3.1.1. Trend Indicators

This subset of indicators evaluates the trend/direction in which the market is moving. They are oscillatory, i.e., tend to move between high and low values. Examples include Exponential Moving Average (EMA), Ichimoku indicator, and Moving Average Convergence Divergence (MACD).

3.1.2. Momentum Indicators

Momentum indicators determine the trend's strength, and signal whether there is any likelihood of reversal. It gives an idea of how momentum develops on a specific currency pair. When the price goes higher, oscillators will move higher, and vice versa. When an oscillator reaches a saturation level, it might indicate a reversal. Examples include the Relative Strength Index (RSI) and Kaufman's Adaptive Moving Average (KAMA).

3.1.3. Volatility Indicators

Volatility is one of the essential indicators for measuring the price change in a given period. It is a two-sided coin. On one hand, volatility is how traders can turn a profit, especially on short-term trades. On the other hand, high volatility indicated less certainty about the market's movements; hence, the price can move in a direction one did not anticipate. Examples include Bollinger Bands and Average True Range (ATR).

3.1.4. Volume Indicators

The volume refers to the number of shares traded for a specific asset. These indicators help to measure the strength of a trend and its direction. When the price changes, volume indicates how strong the move is. Examples include On-Balance volume and Volume Weighted Average Price (VWAP).

3.2 Combination of Indicators

The right combination can strengthen a trader's conviction. Predictor variables that are highly correlated provide little independent explanatory ability. This pattern is known as multicollinearity or simply collinearity. Collinearity increases the variances of the regression coefficient so that they may have theoretically implausible magnitudes or signs, they also vary substantially with minor changes in the sample of observations or the set of predictors and can be individually nonsignificant even though they explain significant amounts of variance overall in Franke et al. (2010).
Hence, similar indicators should not be considered to avoid redundancy, and one should select complementary indicators that do not give redundant results. The right way to achieve this is by picking indicators from each type mentioned above (Momentum, Volume, Volatility, and Trend).
In this paper, we tried multiple combinations from these four groups and found that the below four indicators worked best and gave the most accurate results.

3.2.1. Average True Range (ATR)

It is a market volatility indicator. It is generally derived from the 14-day simple moving average of a series of accurate range indicators. It does not predict or indicate the price movement; instead, it primarily measures volatility caused by gaps and up and down moves. First, the actual range is calculated, determining a share's price range on a given day. Conclusively, the average of absolute degrees for N trading days is calculated.

$$\mathrm{TR}=\mathrm{Max}[(H\text{-}L),\ \mathrm{Abs}\ (H\text{-}C),\ \mathrm{Abs}(L\text{-}C)] \tag{1}$$

$$\mathrm{ATR}=(1/n)\ {}^{(n)}\textstyle\sum_{(i=1)}\ (TR) \tag{2}$$

Where: TR= A particular true range
n= Time period
H= High Price
L=Low Price
C=Close Price

3.2.2. Volume Weighted Average Price

It is a market volume indicator commonly used by short-term traders. It represents the average of a security's price action throughout the day. It is calculated using the volume and price of a security. Firstly, the average price needs to be calculated, which can be done by dividing the sum of High, Low & Close for a particular day by three. The product of the obtained value of typical price and Volume is divided by Cumulative volume to evaluate the value for VWAP.

$$\text{Volume Weighted Average Price} = \frac{Typical\ Price * Volume}{Cumulative\ Volume} \tag{3}$$

Where:

Typical Price = (High price + Low price + Closing Price)/3

Cumulative = total since the trading session opened.

3.2.3. Relative Strength Indicator

It is a momentum indicator that indicates the magnitude of recent price changes to detect overbought or oversold conditions for security. The RSI is essentially an oscillator ranging from 0 to 100.

RSI values of 70 or above suggest that a security is overbought or overvalued and may prime for a trend reversal or corrective pullback in price. Similarly, An RSI value of 30 or below indicates an oversold or undervalued condition.

Initially, RS (Relative Strength) is calculated by dividing the average of the days Closed Up by the average of the days Closed down.

To calculate the final RSI, firstly, the ratio of the calculation is indexed to 100 to ascertain the absolute value. Secondly, average of gain and loss is computed by taking the aggregate of gain and loss and dividing it by the period 'N'. The average of each is then multiplied by 'N-1', and then summed up with their successive values.

Step 1:

$$RS = \frac{Average\ gain}{Average\ loss}$$

$$RSI\ for\ first\ day = 100 - \left(\frac{100}{1+RS}\right) \tag{4}$$

The Average is calculated for N trading days

Step 2:

$$RSI = 100 - \left(\frac{100}{1 + \frac{(Previous\ average\ gain * (N-1) + current\ gain}{(Previous\ average\ loss * (N-1) + current\ loss}}\right) \tag{5}$$

3.2.4. Weighted Moving Average (WMA)

Moving averages are trend indicators. It shows the average price movement of a security over a certain period. The primary difference between simple moving average, and weighted moving average, is the sensitivity toward changes in the data used. SMA gives the average price over a specific period. On the other hand, in WMA, the closed price is weighted with a particular value. The highest weight is assigned to the most recent data, and the oldest data is given a weight of 1. To compute the absolute value sum of calculated values is divided by the sum of weights.

$$\text{WMA} = \frac{(Price1*\text{n} + Price2*(\text{n}-1)+\cdots\ldots Price\text{n}))*2}{n*(n+1)} \tag{6}$$

Where: n= Time period

3.2. Performance Metrics

Whenever we train a model, it is essential to determine the accuracy of the postulated output(s). Performance metrics use different statistics and mathematical equations that compute the differences in actual and predicted values, and determine the accuracy of the trained model. Some of the key performance metrics are described below:

3.3.1 Root Mean Squared Error (RMSE)

This is one of the most common metrics used to evaluate the performance of a regression model. The metric determines the average squared difference between actual and predicted values. The lower the RMSE, the better the model performance is.

$$\text{RMSE} = \sqrt{(\sum(p-o)^2/N)} \tag{7}$$

Where: p= predicted value
o = observed value
N= Time Period

3.3.2 *R Squared Score (R^2 Score)*

R Squared score, also known as the coefficient of determination compares the output values against the mean line of input data. The metric's value lies between 0 & 1, and weight above 0.5 for a model is considered acceptable.

$$\text{SSR} = \sum_i (y_i - \text{f}_i)^2$$
$$\text{SST} = \sum_i (y_\text{i} - \text{y}_\text{mean})^2$$

$$\text{R2} = 1 - \frac{\text{sum squared regression (SSR)}}{\text{(SSR)total sum of squares (SST)}} \tag{8}$$

Where: y_i = i^{th} observation for actual values
f_i = i^{th} observation for predicted values
y_{mean} = mean of actual data

3.3.3 Mean Absolute Percentage Error (MAPE):

As the name suggests, metric computes the mean of absolute percentage error of input values. 0.0 is considered the best value for MAPE. The higher the value, the less accurate the model is.

$$\text{MAPE}= (1/n) \sum_{k=0}^{n} abs(a-p)/a \tag{9}$$

Where: a = actual value
p = predicted value
n = time period

3.3.4 *Mean Absolute Error (MAE)*

The metric firstly calculates the difference between actual and predicted values, takes their absolutes, and then computes the mean of absolute errors. MAE is a domain-specific metric. Thus, the excellent value of MAE depends on the input ingested into the model.

$$\text{MAE} =(1/n) * \sum|p_i - a_i|. \tag{10}$$

Where: p_i = predicted value for i^{th} observation
a_i = actual value for i^{th} observation
n= number of observations

3.4. Neural Network Models

This paper utilizes five neural network/ deep learning models to perform and compare them for time series forecasting.

3.4.1. Long Short-Term Memory (LSTM)

It is a recurrent neural network (RNN) introduced in Hocheiter et al. (1997). It is capable of learning long-term dependencies, especially in sequence prediction problems. LSTM has feedback connections, i.e., it can process the entire data sequence, apart from single data points. It has led to a transformation in machine learning and neurocomputing fields. One of the reasons for the success of LSTM is its ability to handle the vanishing gradient problem, which was earlier difficult to circumvent when training recurrent neural networks proposed in Lee TK et al. (2018).
It is a compelling time-series model that can predict an arbitrary number of steps into the future. An LSTM module (or cell) has five essential components that allow it to model long-term and short-term data.

- Cell state: It is the internal memory of the cell used to store both short-term and long-term memories. It remembers values over arbitrary time intervals. It is responsible for keeping track of the dependencies between the elements in the input sequence.

- Hidden state: It represents the output state information calculated with the current input, previously hidden state, and current cell input used to predict the future stock market prices. Additionally, the secret state can decide only to retrieve both types of memory stored and make the following prediction.

• Input gate: Decides how much information from current input flows to the cell state.
• Forget gate: Decides how much information from the current input and the previous cell state flows into the current cell state.

• Output gate: Decides how much information from the current cell state flows into the hidden state so that, if needed, LSTM can only pick the long-term or short-term memories.

3.4.2. Neural Basis Expansion Analysis for Interpretable Time Series Forecasting(N-BEATS)

N-BEATS stands for Neural basis expansion analysis for interpretable time series forecasting , BEATS is a type of neural network first described in Oreshkin BN et al (2019). The authors reported that N-BEATS outperformed the M4 forecast competition's winner by 3%. The M4 winner model was a hybrid between an RNN and Holt-Winters exponential smoothing — whereas N-BEATS implements a "pure" deep neural architecture. The paper on N-BEATS focused on the univariate time series forecasting problem using deep learning. However, our paper extrapolates the original N-BEATS architecture to multivariate time series by flattening the source data to a 1-dimensional series.
This model consists of a sequence of stacks that combine multiple blocks. The blocks connect feedforward networks via forecast and back cast links. Each block generates partial estimates, focusing on the local characteristics of the time series, and causes residual errors that are not disentangled. The stacks' purpose is to identify non-local patterns along the complete time axis by "looking back." Finally, the partial forecasts are pieced together into a global forecast at the model level.

3.4.3. Temporal Fusion Transformer (TFT)

TFT is an attention-based Neural Network optimized for outstanding performance and forecasting. Lim B et al. (2021) reported various advantages, including rich feature support, high performance, and interpretability, and it also provides support to heterogeneous time-series and multi-horizon forecasting.
TFT model stands on a complex architecture, mainly consisting of: LSTM Blocks that identify the relationship of time steps with surrounding values, variable selection networks used to select input at each stage, Gated Relation Networks (GRN) whose role is to discard unwanted inputs and prevents overfitting, static covariate encoders incorporate static features within the network, and attention blocks that encapsulate long-term dependencies and recognize most pertinent patterns.

3.4.4. Gated Recurrent Unit (GRU)

GRU was first introduced by Cho et al. (2014), designed to solve the vanishing gradient problem that has been an issue in a standard recurrent neural network. GRU is also considered a simplified version of the LSTM model, using only one state vector and two gated vectors. It can retain long-term dependencies with almost the same accuracy as LSTM, with the advantage of superior speed. GRUs simplified structure figures out the dependencies from initials parts of the data without even neglecting them; this is done using gating units which also determine if information regulated is vital in output prediction and needs to be kept or discarded.

The structure mainly consists of three gates. Update gate(z) determines how much information from previous time steps needs to be fed along with the future data, whereas reset gate(r) determines the amount of information from the past that can be discarded, and the current memory gate adds some non-linearity to the input, also, it reduces the intensity of past information on existing data.

3.4.5. Temporal Convolution Network (TCN)

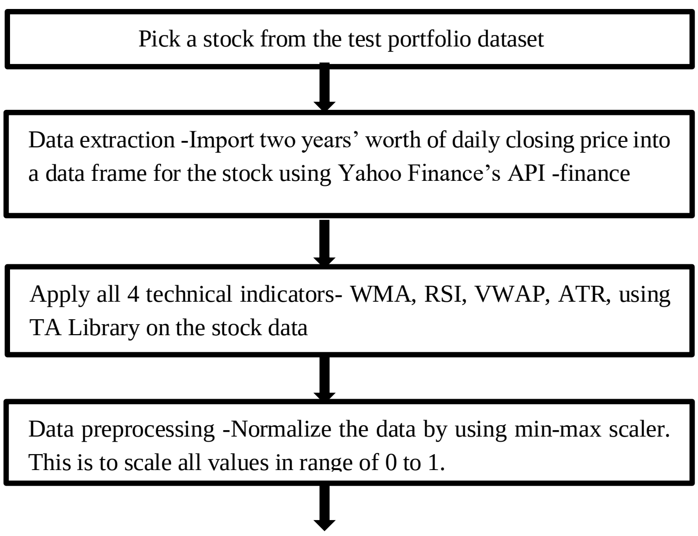
The model was first proposed by Lea et al. (2016) for video-based action segmentation. It combines features of RNN and CNN into a single architecture. CNN's one dimension layer job is encoding low-level features using spatial-temporal information, whereas RNN catches high-level temporal information.
Convolutions are designed to be causal, so no information leakage happens in the architecture. The model also maps the input sequence to same length output, thus fulfilling role of RNNs.
The architecture, consists of two parts, encoders, and decoders, each of them is composed of temporal convolutions, 1D pooling/upsampling, and channel-wise normalization layers. Convolution layers within the encoder are used to filter in required input. The second layer is the pooling layer, whose role is to evaluate activations efficiently for the long haul. The channel-wise normalization layer further normalizes the information. The decoder performs akin operations barring the pooling layer part, which is replaced by upsampling, that firstly up samples the values, convolute them, and then conducts normalization.

4 Experiment

4.1 Datasets

As stock market prices fluctuate and are dynamic, we have utilized only two years of daily closing price data from Yahoo Finance API for short-term price prediction. We randomly picked up two shares from five sectors listed on NSE: Auto, Bank, Oil and Gas, Metal, and Fast-moving consumer goods (FMCG). Picking shares from five different sectors helped in achieving portfolio diversity. Moreover, it also helped in testing our models' robustness and versatility.

4.2 Methodology



Divide the data into Training (80 %), Validation (10%), and Testing Set (10%)

↓

Before, training each model, tune the models with appropriate parameters, covariates and hyperparameters including but not limited to no of epochs (here 300), covariates (here normalized technical indicators), batch size, hidden layer etc. Train and test each model on the stock data and evaluate all four performance metrics as mentioned before for each of them.

↓

Plot the actual and predicted closing price predicted for each model.

Flow Chart 1: The flow chart interprets the working of the code(s) developed for research analysis

4.3 Experimental Results

4.3.1. Nifty Auto Sector

It is evident from Table 1 that LSTM outperformed for BAJAJ AUTO stock as it has the slightest error in terms of RMSE and MAE, and a R^2 score close to 1(0.95) and a low MAPE. It is followed by TFT, which has comparable performance with LSTM.GRU also depicted a moderate accuracy, whereas TCN and N-BEATS had the worst performance. This can also be visually confirmed from fig 1-5 .

Similarly, LSTM outperformed for BOSCHLTD stock with the slightest error in terms of RMSE, MAE, and R^2 score close to 1, and a low MAPE. It is followed by GRU, which has comparable performance with LSTM.TFT also depicted a moderate accuracy, whereas TCN and N-BEATS again had the worst performance with very high RMSE, MAPE, MAE, and a negative R^2 score. LSTM accuracy can be visually confirmed from fig 6-10.

Table 1
Performance analysis for Auto Sector

SECTOR	Stock Name	Symbol	Model	Errors			
				RMSE	MAPE	MAE	R2 Score
NIFTY AUTO	Bajaj Auto	BAJAJ-AUTO	**LSTM**	**24.22**	**0.45**	**16.36**	**0.95**
			GRU	39.13	0.88	33.8	0.88
			TCN	99.54	2.42	89.9	-6.55
			TFT	24.96	0.63	23.2	0.94
			N-BEATS	131.85	2.84	105.58	-1.36
	Bosch Limited	BOSCHLTD	**LSTM**	**177.98**	**1.02**	**149.24**	**0.96**
			GRU	208.42	1.25	180.17	0.95
			TCN	505.73	3.04	446.16	0.32
			TFT	244.42	1.54	223.89	0.91
			N-BEATS	841.84	5.01	724.32	-2.47

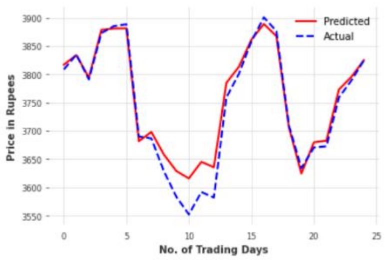


Figure 1: LSTM model for Bajaj

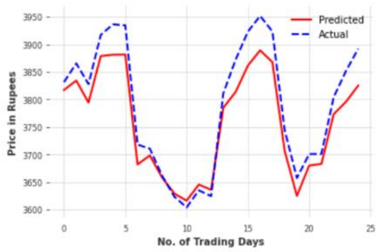


Figure 2: GRU model for Bajaj

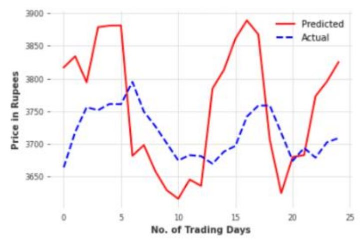


Figure 3: TCN model for Bajaj

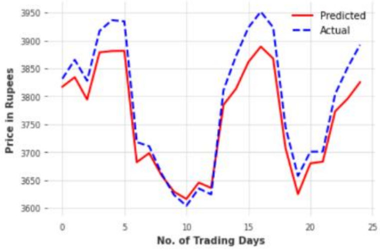


Figure 4: TFT model for Bajaj

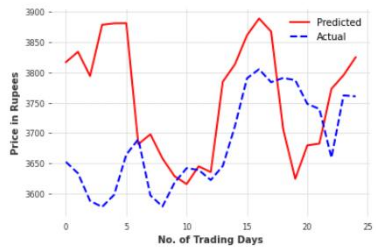


Figure 5: N-BEATS model for Bajaj

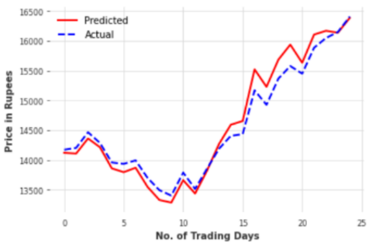


Figure 6: LSTM model for Bosch

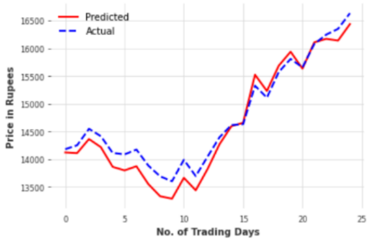


Figure 7: GRU model for Bosch

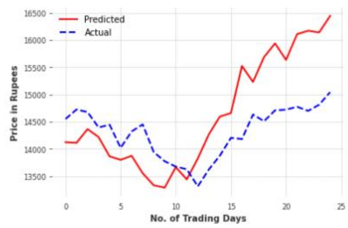


Figure 8: TCN model for Bosch

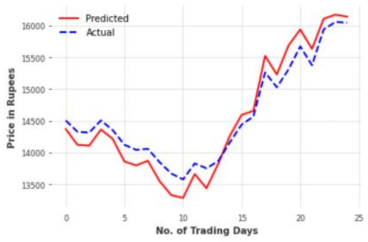


Figure 9: TFT model for Bosch

Figure 10: N-BEATS model for Bosch

4.3.2 Nifty Bank Sector

It is evident from Table 2 that GRU outperformed in HDFC bank stock prediction as it has the slightest error in terms of RMSE, MAE, and MAPE, and an optimal value of R^2 Score, moreover, LSTM was the second-best performer. TFT also depicted a moderate accuracy, whereas TCN and N-BEATS had the worst performance with very high RMSE, MAPE, MAE, and negative R^2 scores. This can also be visually confirmed from fig 11-15, where GRU and LSTM predicted the closing price for 25 trading days with the most accuracy.

Similarly, GRU outperformed in ICICI bank stock prediction as it has the slightest error regarding RMSE, MAE, and MAPE, and a perfect R^2 Score of 1. It is followed by TFT. LSTM also depicted a moderate accuracy, whereas TCN and N-BEATS had the worst performance with very high RMSE, MAPE, MAE, and a low R^2 score. The same can be visualized in fig 16-20.

Table 2
Performance analysis of the Bank Sector

SECTOR	Stock Name	Symbol	Model	Errors			
				RMSE	MAPE	MAE	R2 Score
NIFTY BANK	HDFC Bank	HDFCBANK	LSTM	10.81	0.71	9.4	0.88
			GRU	**6.52**	**0.38**	**5.08**	**0.93**
			TCN	130.61	5.32	122.89	-4.96
			TFT	37.28	1.56	34.45	0.45
			N-BEATS	79.02	2.93	65.96	-1.34
	ICICI Bank	ICICI	LSTM	4.25	0.52	3.72	0.96
			GRU	**1.46**	**0.17**	**1.22**	**1**
			TCN	14.37	1.64	11.78	0.39
			TFT	3.51	0.42	3.01	0.97
			N-BEATS	19.79	2.25	16.32	-0.37

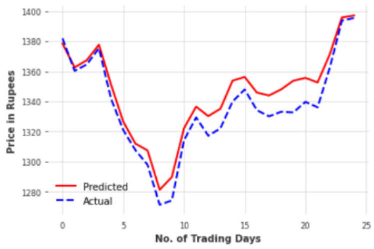


Figure 11: LSTM model for HDFC

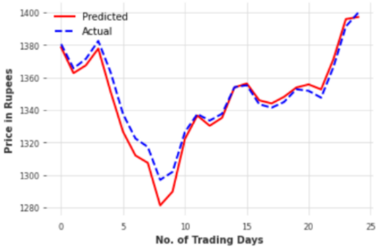


Figure 12: GRU model for HDFC

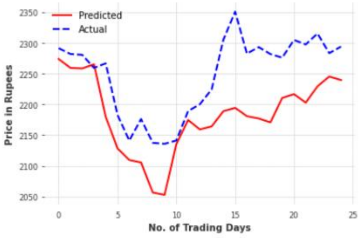


Figure 13: TCN model for HDFC

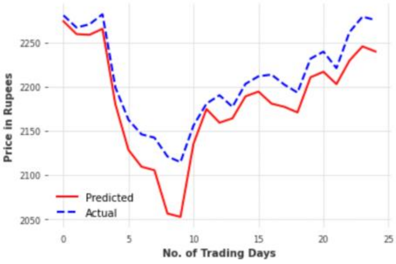


Figure 14: TFT model for HDFC

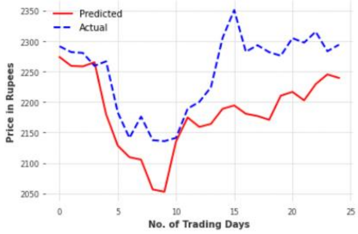


Figure 15: N-BEATS model for HDFC

Figure 16: LSTM model for ICICI

Figure 17: GRU model for ICICI

Figure 18: TCN model for ICICI

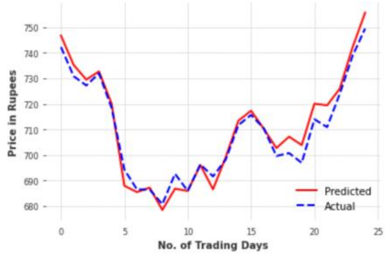


Figure 19: TFT model for ICICI

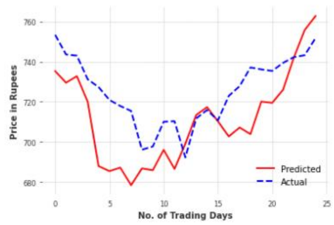


Figure 20: N-BEATS model for ICICI

4.2.3. Nifty Metal Sector

As interpreted in Table 3 ,GRU outperformed in price prediction for stock ADANIENT, as it has the slightest error, and R^2 score close to 1. The performance of LSTM is comparable to the GRU. TFT & TCN performance was abysmal compared to LSTM & GRU, whereas N-BEATS proved to be the worst model again for the share with very high values for RMSE, MAPE, and MAE, and a negative R^2 score. Trends in fig 21-25, represents GRU to be the most accurate model for the share, and TCN & N-BEATS output values failed to follow the trend.

For WELCORP stock, both GRU and LSTM outshines every other model with the least errors, and highest R^2 score. Surprisingly, N-BEATS and TCN performed better than TFT. TCN interpretation seems similar to the N-BEATS in all metrics except the R^2 score. TFT showed very poor performance for the share.
Overall, GRU is the winner in the NIFTY metal sector. The same can be interpreted from fig 26-30, where LSTM and GRU models' predicted values have shown the highest accuracy.

Table 3
Performance analysis for Metal Sector

SECTOR	Stock Name	Symbol	Model	Errors			
				RMSE	MAPE	MAE	R2 Score
NIFTY METAL	Adani Enterprises	ADANIENT	LSTM	15.38	0.54	11.7	0.93
			GRU	**13.59**	**0.51**	**10.97**	**0.95**
			TCN	58.52	2.37	51.36	-4.06
			TFT	40.67	1.66	35.95	0.18
			N-BEATS	652.39	42.42	650.12	-122.3
	Welspun Corp Limited	WELCORP	**LSTM**	**4.48**	**1.78**	**3.75**	**0.81**
			GRU	**4.57**	**1.73**	**3.67**	**0.84**
			TCN	9.87	4.04	8.58	-0.74
			TFT	18.25	8.89	17.22	-0.14
			N-BEATS	9.23	3.47	7.57	-0.32

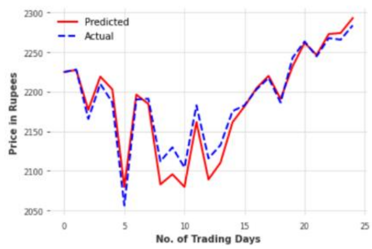


Figure 21: LSTM model for Adani

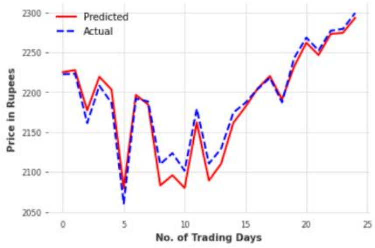


Figure 22: GRU model for Adani

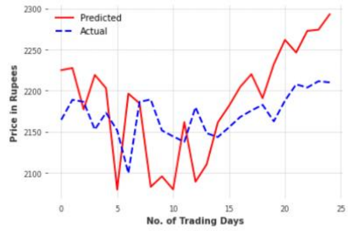


Figure 23: TCN model for Adani

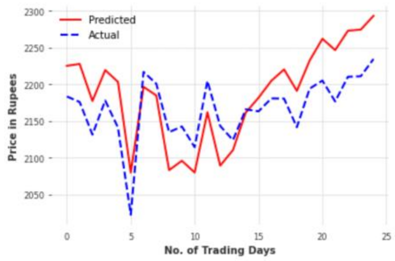


Figure 24: TFT model for Adani

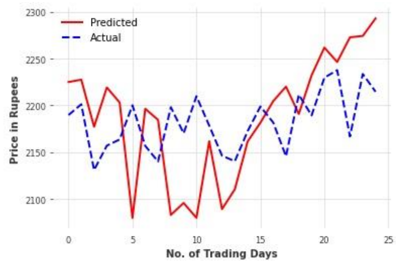


Figure 25: N-BEATS model for Adani

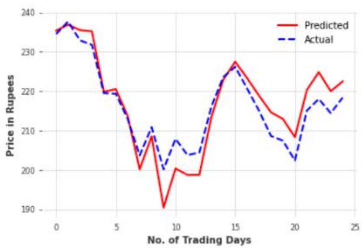


Figure 26: LSTM model for Welspurn

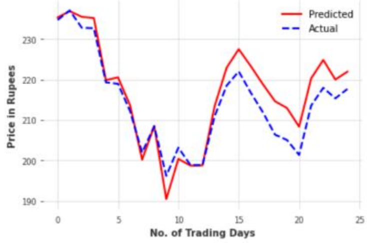


Figure 27: GRU model for Welspurn

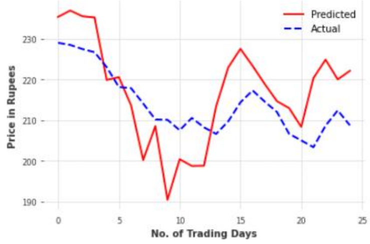


Figure 28: TCN model for Welspurn

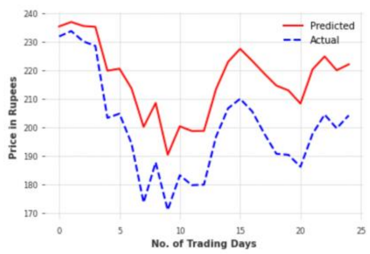


Figure 29: TFT model for Welspurn

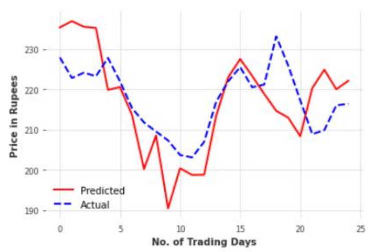


Figure 30: N-BEATS model for Welspurn

4.2.4. Nifty FMCG Sector

In Table 4, GRU outperformed again, for the Britannia stock as it has the least error and highest R^2 score of 0.9 compared to other models. TFT stands out as the second-best performing model, whereas the performance of LSTM was modest for the share. TCN & N-BEATS performance is the poorest for the stock. This can also be visually confirmed from fig 31-35, where GRU and TFT predicted the closing price for 25 trading days with the most accuracy.

For, MARICO stock TFT performed best with the least RMSE, MAPE, and MAE, and an R^2 score of 0.89, followed by GRU as the second-best performer. LSTM performed modestly and TCN & N-BEATS again failed to perform even decently, fig 36-40, visualizes the same.

Table 4

Performance metrics for FMCG sector

SECTOR	Stock Name	Symbol	Model	Errors			
				RMSE	MAPE	MAE	R2 Score
NIFTY FMCG	Britannia Industries Limited	BRITANNIA	LSTM	64.45	1.49	52.08	0.73
			GRU	**38.68**	**0.81**	**28.66**	**0.9**
			TCN	132.6	2.87	-3.58	99.69
			TFT	43.64	0.86	30.71	0.85
			N-BEATS	135.6	2.89	101.24	-4.69
	Marico Limited	MARICO	LSTM	5.08	0.64	3.22	0.84
			GRU	3.98	0.66	3.22	0.9
			TCN	11.47	1.9	9.98	-1
			TFT	**3.73**	**0.54**	**2.71**	**0.89**
			N-BEATS	28.66	4.9	25.4	-10.23

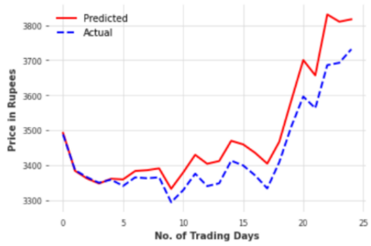


Figure 31: LSTM model for Britannia

Figure 32: GRU model for Britannia

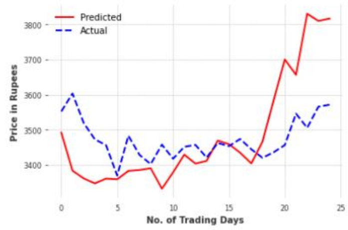


Figure 33: TCN model for Britannia

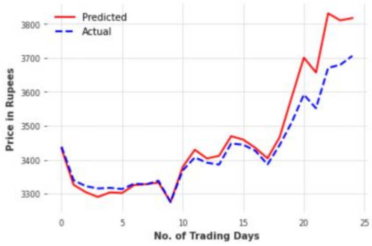


Figure 34: TFT model for Britannia

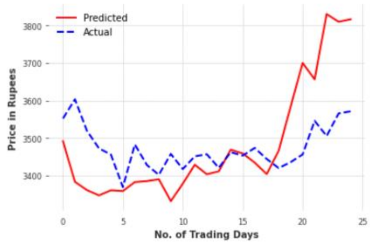


Figure 35: N-BEATS model for Britannia

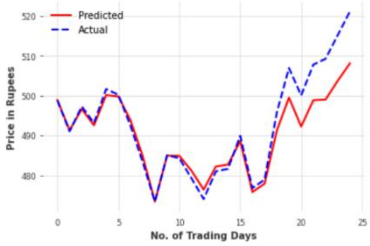


Figure 36: LSTM model for Marico

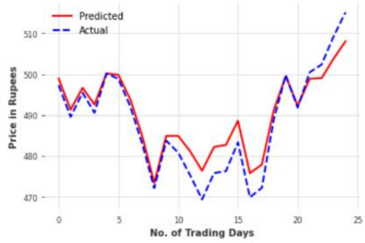


Figure 37: GRU model for Marico

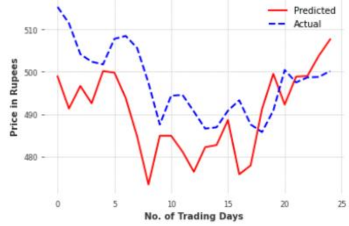


Figure 38: TCN model for Marico

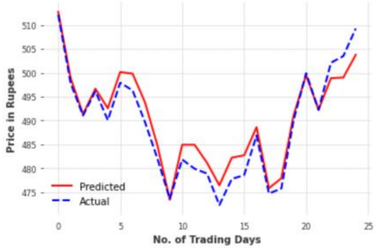


Figure 39: TFT model for Marico

Figure 40: N-BEATS model for Marico

4.2.5. Nifty Oil & Gas Sector

In Table 5, LSTM outperformed in RELIANCE stock with the least RMSE, MAE, and MAPE, and an excellent R^2 score of 0.99, whereas GRU has done a comparable job with an equal R^2 Score of 0.99. TFT again stands out as a modest performer, however, TCN & N-BEATS again showed the worst performance. LSTM and GRU follows the trends precisely and the same can be seen in fig 41-45.

In the case of IGL, TFT outperformed all the models with the least RMSE, MAE, and MAPE, and an R^2 score of 0.92. LSTM had a decent performance, surprisingly GRU had a poor performance with a negative R^2 score. TCN and N-BEATS were again the underperformers. The fig 46-50 represents TFT & LSTM models' predicted values are the closest to the actual values.

Table 5
Performance metrics for the Oil & Gas sector

SECTOR	Stock Name	Symbol	Model	Errors			
				RMSE	MAPE	MAE	R2 Score
NIFTY OIL & GAS	Reliance Industries Limited	RELIANCE	**LSTM**	**11.27**	**0.35**	**8.85**	**0.99**
			GRU	12.32	0.4	10.18	0.99
			TCN	78.65	2.54	0.04	65.76
			TFT	38.81	1.34	34.29	0.88
			N-BEATS	86	2.51	65.23	0.32
	Indraprastha Gas Limited	IGL	LSTM	4.62	1.12	4.03	0.63
			GRU	8.3	2.15	7.58	-0.2
			TCN	9.76	2.23	7.85	-1.47
			TFT	**2.78**	**0.66**	**2.37**	**0.92**
			N-BEATS	13.88	3.29	12.12	-1.42

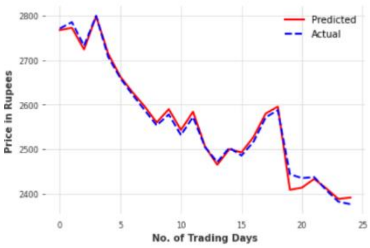


Figure 41: LSTM model for Reliance

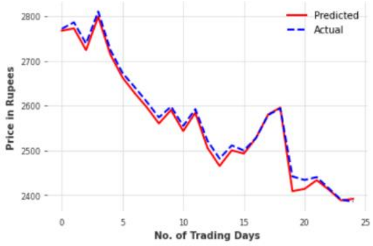


Figure 42: GRU model for Reliance

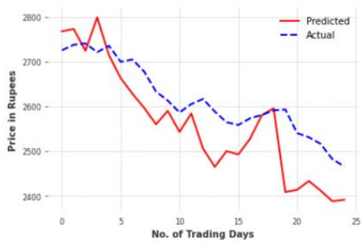


Figure 43: TCN model for Reliance

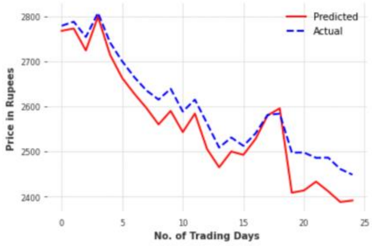


Figure 44: TFT model for Reliance

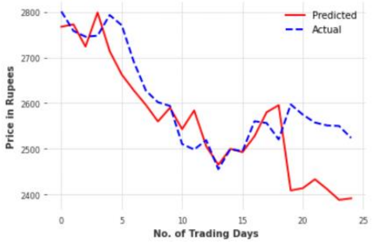


Figure 45: N-BEATS model for Reliance

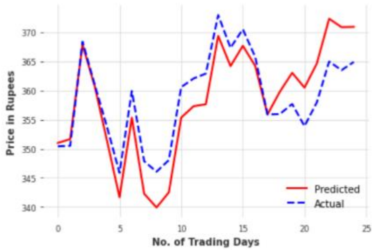


Figure 46: LSTM model for IGL

Figure 47: GRU model for IGL

Figure 48: TCN model for IGL

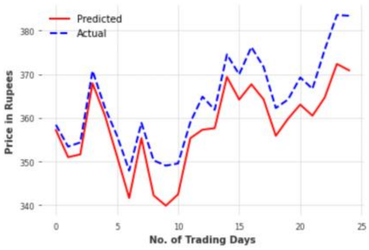


Figure 49: TFT model for IGL

Figure 50: N-BEATS model for IGL

4 Conclusion

Deep learning has revolutionized the landscape of time series forecasting. The purpose of this paper was to do an empirical analysis of different neural networks/deep learning models on diverse test portfolio from stocks listed on NSE. Out of 10 stocks, GRU outperformed in 4, LSTM outperformed in 3 stocks and both GRU and LSTM had similar performance in 1 stock. Moreover, TFT outperformed in remaining 2 stocks. In other words, RNN models- GRU and

LSTM outperformed the newer models- TFT, TCN, and N-BEATS. TFT performed better than TCN and N-BEATS, and in some cases had comparable performance to GRU and LSTM. The dire performance in N-BEATS, TCN, and sub-par performance in TFT can be attributed to relatively less community-wide research and investment into architectural and algorithmic elaborations, as seen in older models like LSTM. In the future, we would like to explore, combine and use these new state-of-the-art models for other time-series forecasting problems.

Funding No funding was received to assist with the preparation of this manuscript.

Data availability The datasets generated during and/or analysed during the current study are fetched from the site of yahoo finance, https://finance.yahoo.com/ .

Declarations

Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors**.** This research is solely the work of the authors and their employer organization has nothing to do with the research performed. Moreover, data and content used for research has no relevance to the organization.

References

Althelaya KA, El-Alfy E-SM, Mohammed S (2018) Stock Market Forecast Using Multivariate Analysis with Bidirectional and Stacked (LSTM, GRU),2018 21st Saudi Computer Society National Computer Conference (NCC),pp. 1-7, DOI: 10.1109/NCG.2018.8593076.

Ariyo AA, Adewumi AA, Ayo CK (2014) Stock Price Prediction Using the ARIMA Model, UKSim-AMSS 16th International Conference on Computer Modelling and Simulation, pp. 106-112, DOI: 10.1109/UKSim.2014.67

Bai S, Kolter JZ, Koltun V (2018) An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence, volume= {abs/1803.01271} arXiv preprint DOI:10.48550/arXiv.1803.0127

Banerjee D (2014) Forecasting of Indian stock market using time-series ARIMA model, 2014 2nd International Conference on Business and Information Management (ICBIM), pp. 131-135, DOI: 10.1109/ICBIM.2014.6970973.

Cho K, Merrienboer BV, Bahdanau D, Bengio Y (2014). On the Properties of Neural Machine Translation: Encoder-Decoder Approaches arXiv preprint DOI:10.48550/arXiv.1409.1259

Franke GR (2010) Multicollinearity, Part 2. Marketing Research, Wiley International Encyclopaedia of Marketing, 2010, DOI:10.1002/9781444316568.wiem02066xx

Greco M, Spagnoletta M, Appice A, Malerba D (2021). Applying Machine Learning to Predict Closing Prices in Stock Market: A Case Study. In: Bitetta V, Bordino I, Ferretti A, Gullo F, Ponti G, Severini L (eds) Mining Data for Financial Applications. MIDAS 2020. Lecture Notes in Computer Science(), vol 12591. Springer, Cham. DOI: 10.1007/978-3-030-66981-2_3

Hochreiter S, Schmidhuber J (1997) "Long Short-Term Memory," in Neural Computation, vol. 9, no. 8, pp. 1735-1780, 15 Nov. 1997, DOI:10.1162/neco.1997.9.8.1735.

Hu X (2021) Stock Price Prediction Based on Temporal Fusion Transformer, 3rd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI), pp. 60-66, DOI: 10.1109/MLBDBI54094.2021.00019.

Lea C, Vidal R, Reiter A, Hager GD (2016) Temporal Convolutional Networks: A Unified Approach to Action Segmentation. In: Hua G., Jégou H. (eds) Computer Vision – ECCV 2016 Workshops. ECCV 2016. Lecture Notes in Computer Science(), vol 9915. Springer, Cham. DOI:10.1007/978-3-319-49409-8_7

Lee TK, Cho JH, Kwon DS, Sohn SY (2019) Global stock market investment strategies based on financial network indicators using machine learning techniques, Expert Systems with Applications, Volume117,Pages 228-242,ISSN 0957-4174,DOI: 10.1016/j.eswa.2018.09.005.

Lim B, Arık SÖ., Loeff N, Pfister T (2021) Temporal Fusion Transformers for interpretable multi-horizon time series forecasting, International Journal of Forecasting, Volume37, Issue4, Pages1748-1764,ISSN0169-2070,DOI: 10.1016/j.ijforecast.2021.03.012.

Liu M, Taylor JW, Choo W-C (2020) Further empirical evidence on the forecasting of volatility with smooth transition exponential smoothing, Economic Modelling, Volume93, Pages,651-659,ISSN,02649993,DOI: 10.1016/j.econmod.2020.02.021.

Liu Y, Dong H, Wang X, Han S (2019) Time Series Prediction Based on Temporal Convolutional Network, 2019 IEEE/ACIS 18th International Conference on Computer and Information Science (ICIS), pp. 300-305, DOI: 10.1109/ICIS46139.2019.8940265.

M H., E.A. G., Menon VK, and K.P., S. (2018) NSE Stock Market Prediction Using Deep-Learning Models, Procedia Computer Science, Volume 132, Pages 1351-1362, ISSN 1877-0509, DOI: 10.1016/j.procs.2018.05.050.

Nabipour M, Nayyeri P, Jabani H, S. S. and Mosavi M (2020) Predicting Stock Market Trends Using Machine Learning and Deep Learning Algorithms Via Continuous and Binary Data; a Comparative Analysis, in IEEE Access, vol. 8, pp. 150199-150212, DOI: 10.1109/ACCESS.2020.3015966.

Nti IK, Adekoya AF, Weyori BA (2020) A systematic review of fundamental and technical analysis of stock market predictions. Artif Intell Rev 53, 3007–3057. DOI:10.1007/s10462-019-09754-z

Oreshkin BN, Carpov D, Chapados N, Bengio Y (2019) N-BEATS: Neural basis expansion analysis for interpretable time series forecasting, arXiv preprint DOI:10.48550/arXiv.1905.10437

Oriani FB, and Coelho GP (2016) Evaluating the impact of technical indicators on stock forecasting,2016 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1-8, DOI:10.1109/SSCI.2016.7850017.

Peng Y, Albuquerque PHM, Kimura H, Saavedra CAPB (2021) Feature selection and deep neural networks for stock price direction forecasting using technical analysis indicators, Machine Learning with Applications,Volume5,100060, ISSN 2666-8270, DOI:10.1016/j.mlwa.2021.100060.

Samarawickrama AJP, Fernando TGI (2017) A recurrent neural network approach in predicting daily stock prices an application to the Sri Lankan stock market, IEEE International Conference on Industrial and Information Systems (ICES), pp. 1-6, DOI: 10.1109/ICIINFS.2017.8300345

Sbrana A, Rossi ALD, Naldi MC (2020) N-BEATS-RNN: deep learning for time series forecasting, 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 765-768, DOI: 10.1109/ICMLA51294.2020.00125.

Sharma A, Bhuriya D, Singh U (2017) Survey of stock market prediction using machine learning approach, international conference of Electronics, Communication, and Aerospace Technology (ICECA), pp. 506-509, DOI: 10.1109/ICECA.2017.8212715.

Wu B, Wang L, Zeng Y-R (2022) Interpretable wind speed prediction with multivariate time series and temporal fusion transformers, Energy, Volume 252, 123990, ISSN 0360-5442, DOI: 10.1016/j.energy.2022.123990

Zhang H, Zou Y, Yang X, Yang H (2022) A temporal fusion transformer for short-term freeway traffic speed multistep prediction, Neurocomputing, Volume 500, Pages 329-340, ISSN 0925-2312, DOI: 10.1016/j.neucom.2022.05.083.

Notes

Investopedia, Finance Media Website,2022. [Online]. Available: https://www.investopedia.com/

NSE, National Stock Exchange, 2015. [Online]. Available: http://www.twse.com.tw/

TA-Lib, TA-Lib: Technical Analysis Library,2015. [Online]. Available: http://ta-lib.org/

Yahoo Finance, Yahoo finance - business finance, stock market, quotes, and news, 2015. [Online]. Available: http://finance.yahoo.com/