# Application of LSTM, GRU and ICA for Stock Price Prediction: Proceedings of ICTIS 2018, Volume 2

**Chapter** · January 2019

DOI: 10.1007/978-981-13-1747-7_46



Purva Raut
Dwarkadas J. Sanghvi College of Engineering

# Application of LSTM, GRU & ICA for Stock Price Prediction

Mr. Akhil Sethia[1[0000-1111-2222-3333]] and Prof. Purva Raut[2[1111-2222-3333-4444]]

[1] Dwarkadas J. Sanghvi College of Engineering, Mumbai, India

[2] Assistant professor, Dwarkadas J. Sanghvi College of Engineering, Mumbai, India

**Abstract.** This paper attempts to provide an optimal model for the prediction of stock prices for t+5$^{th}$ day and consequently provide a daily buying/selling strategy for the Standard's & Poor's 500 Index. A performance comparison between LSTM, GRU, ANN and SVM model has been made and an optimal model has been outlined. Training and prediction data spanned over 12 years from 2000 to 2017. 50 technical indicators based attributes were calculated and appended to the Open, High, Low, Close & Volume (OHLCV) data for S&P500, each attribute value was converted into a relative standard score followed by Minimax scaling and dimensionality reduction, through ICA. The performance of the different models on this dataset were then compared using self defined metrics like optimism and pessimism ratio and returns ratio. The LSTM model proved to outperform the other models with a return of 400% greater than the hold and wait strategy and R2 score of 0.9486.

**Keywords:** Long Short Term Memory (LSTM), Gated Recurrent Unit (GRU), Artificial Neural Networks (ANN), Support Vector Machines (SVM), Financial Time Series Forecasting, Independent Components Analysis (ICA)

## 1. Introduction

Efficient market hypothesis states that the share price reflects all relevant information about a stock are singularly sufficient for future price analysis. Statistical models like ARIMA and GARCH and machine learning models like Support Vector Machines and Artificial Neural Networks are ubiquitously used for stock price predictions but these models are unable to gauge the sequentiality of a developing price trend. A Long Short Term Memory and a Gated Recurrent Unit model, variants of the recurrent neural network, unlike other methods would back propagate through immediate historic prices along with current price and be more apt is detecting a developing trend.

The aim of this study is to propose an optimal model for stock price prediction and building a trading strategy and compare its performance with SVM and ANN models. The models have been trained for 12 years on the Standard's & Poor's 500 Index data. The initial tick data is appended with technical indicators to filter momentary noise due to price spikes and then ICA has then been used to reduce the dimensionality of this dataset. Self defined metrics like optimism, pessimism and return ratio are then used to record the potential return and the confidence level of predictions by the model. Such a model would not only be applicable for stock prices, but would be pertinent in other time series analysis problems like signal processing.

The domain of price prediction in stocks is a thoroughly studied domain. Kwon and Moon (2007)[1] use a simple recurrent network along with a genetic algorithm for predictions on a dataset based on technical indicators. Teixeira & Oliveira (2010)[6] used nearest neighbour classification on a similar dataset to build a model with similar results but with lower computational costs.

Atsalakis & Valavanis (2009)[4] studied over a 100 papers from 1992 to 2006. All variations of neural networks are seen in this study, but few papers propose LSTM and GRU models. Data pre-processing is generally performed by PCA, Minimax and log transformations. The architectures, training and validation sets and the achieved results have been surveyed. Guresen, Kayakutlu & Daim (2011)[8] discuss DAN2 architecture and compares its effectiveness with a general and ARCH and GARCH hybridised MLP model. The MLP model outperforms the DAN2 model and hybridisation does not have an additional impact on the performance.

Patel, Shah, Thakkar & Kotecha (2014)[3,5] review the use of a 2 staged process where a SVR is used to predict all individual attributes of the $t+n^{th}$ day and the price of stock is now predicted using these attributes.They have also studies use of SVM, ANN, Random forests, Naive Baye's classifier predictions for BSE Sensex, Reliance and Infosys were made. Singh & Srivastava (2016)[2] use 2-Dimensional PCA along with a multilayer perceptron model. Different 2 dimensional windows based on the number of days of forecast and the number of attributes in the dataset are experimented, and an optimal window is found. Chen, Zhou & Dan (2015)[7] use a LSTM model on Chinese Stock market data for making predictions. The training data has been sampled from time periods which provide varying amount of returns.

In all the surveyed studies, LSTM models are used scarcely and very few studies pre-process the data thoroughly. This greatly alters the performance of the model. This study fills that gap. Standardisation on the data improves the performance of the models by centring noise from trend reversal signals and normalisation prevents the model weights from being skewed. This study introduces the use of GRU models and ICA for price predictions and optimises the performance of LSTM models. These techniques have not been used before.
Section 2 discusses the currently established studies and Section 3 outlines the data pre-processing and research. Section 4 compared the observed result for each model and compared them. Section 5 provides a conclusion and scope for further development.

## 2. Data Research

### 2.1 Data Pre-processing

Standard's and Poor's 500 index tick data containing Open, High, Low, Close, Volume and Adjusted Close, sourced from Yahoo finance[13] from 3rd January 2000 till 30th October 2017 has been chosen for this study. The Close attribute, inconsistent in events of stock splits and dividends is dropped and Adjusted Close has been used as the target attribute.

This tick data is appended with different technical indicators, which when used in combination, distinguish between a noisy temporary price spike from a long term trend reversal. This would help our model distinguish between a genuine price trend and a market anomaly. They can be broadly classified into 4 fundamental kinds: volume, trend, momentum and volatility indicators. Volume indicators like OBV, use the trading volume to determine the strength of a continuing trend. Momentum indicators like RSI, Stochastic Indicators find out the rate of change of price in a given period of time and represent the health of a current rally. Trend Indicators like Fibonacci retracements, MACD are used to pick up newly any developing trend reversals within the markets. Volatility indicators like Bollinger Bands & ATR have been used. Daily, weekly and monthly returns of past 8 and 2 trading weeks and months has been calculated. The list of indicators is specified in Table 1.

Table 1 - List of Attributes

| **Attributes** | |
|---|---|
| Open, High, Low, Close, Volume | 14 & 21 Day Moving Average Convergence Divergence |
| Daily Returns | Pivot Point |
| 5-Day Momentum | 14-Day Average True Range |
| 14 Day Simple & Exponential (k=2/5) Moving Average | 14-Day Relative Strength Index |
| 14 Day Bollinger Bands | On Balance Volume |
| 14 Day Fast, Slow & Smoothed Slow Stochastic Indicators | 7, 14 & 21 Day Up & Down Trending Fibonacci Retracement at 38.2, 50 & 61.8% |
| Past 8 Weekly & Past 2 Monthly Returns | 3 Day Rate of Change |

The dataset attributes contain varying range of values, by using Z-score standardisation, lower magnitude values, representing no signal or trend will be suppressed, while higher values will be magnified. This will make weaker trend reversal signals have high magnitude and be explicit. Minimax normalisation is then used on this standardised data to scale values between a range of 0 to 1. A Z-score for each attribute in each record is calculated by the following:

$$Z = \frac{\overline{X} - \mathrm{E}[X]}{\sigma(X)/\sqrt{n}} \qquad (1)$$

The regression parameter for the model is the 'Adjusted Close' value of t-5th day. The training data ranges between 7th March 2000 to 1st May 2012 and has 3053 records. The validation set dates from 2nd May 2012 to 11th June 2014 and has 525 records. The testing set contains records from 12th June 2014 to 23rd October 2017 and has 847 records. Hold out validation is used for training the ANN and recurrent network models. Post splitting, feature extraction is performed on this dataset, to get the optimum number of attributes in the dataset. The dataset contains 48 attributes, of which few may be linearly dependent on each other. To avoid the curse of dimensionality, feature extraction is performed. Feature extraction filters and retains the independent attributes in the data, retaining maximum information within the dataset.

### 2.2 Independent Components Analysis

ICA[14] has been used for dimensionality reduction on this dataset. When D'Agostino's $K^2$ test was conducted, except '14-day Uptrend Fibonacci Retracement at 61.8%' all attributes follow a non-gaussian distribution. The dataset contains indicators which represent fundamentally different aspects of the time series. These 2 properties of, distribution being non-gaussian and have different individual components warrant the use of ICA for this data. Table 2 illustrates the prediction results with different number of components. By experimentation, we choose 12 independent components for our final dataset.

Table 2 - This Experimentation has been performed on an LSTM model with optimal model configurations.

| No. of Components | RMSE | R^2 Score | Return Ratio | Optimism Ratio | Pessimism Ratio |
|---|---|---|---|---|---|
| 7 | 0.000503 | 0.936713 | 2.945463 | 0.246706 | 0.298203 |
| 12 | 0.000553 | 0.930429 | 4.921270 | 0.398802 | 0.176048 |
| 18 | 0.000618 | 0.922229 | 1.988380 | 0.522155 | 0.140119 |
| 25 | 0.000536 | 0.932515 | 1.705043 | 0.334132 | 0.238323 |
| 32 | 0.000470 | 0.940888 | 3.763431 | 0.247904 | 0.273053 |
| 45 | 0.000733 | 0.907813 | 1.070164 | 0.439521 | 0.201197 |

## 3. Models

This study compares variants of recurrent neural network models, LSTM and GRU with ANN and SVM models. Implementation of each model has been discussed below.

### 3.1 Long Short Term Memory

Long Short Term Memory[16] model is a specialised RNN model which eliminates the vanishing gradient problem when handling long term dependencies. This issue is solved by maintaining a cell state '$c_t$', which can store information for a long sequence of time. Each cell state is calculated from the previous cell state and a forget gate '$f_t$' and '$g_t$' gate multiplied to the input gate '$i_t$'. The forget gate defines the degree of old cell state to be passed on to the new cell state. The output gate '$o_t$' defines the degree of the cell state to be passed onto the hidden state and hence, to the recurrent unit. The weights of the input, output and forget gates are calculated using back propagation through time and the current hidden state is then calculated by the output gate and the current cell state as shown in equations 2-7.

$$i_t = \sigma(x_t U^i + h_{t-1} W^i) \quad (2)$$
$$f_t = \sigma(x_t U^f + h_{t-1} W^f) \quad (3)$$
$$o_t = \sigma(x_t U^o + h_{t-1} W^o) \quad (4)$$
$$g_t = \tanh(x_t U^g + h_{t-1} W^g) \quad (5)$$
$$h_t = \tanh(c_t).o \quad (6)$$
$$c_t = c_{t-1}.f + g.i \quad (7)$$

### Gated Recurrent Unit

Gated Recurrent Unit[15] is a modified LSTM model with 2 gates. A reset gate '$r_t$' and an update gate '$z_t$' is used to form the hidden state '$h_t$'. The reset gate determines the how to combine the current input in the state with the historic memory. Update gate is responsible for deciding the degree of historic memory which should be maintained in the node. Back propagation through time is used to train the weights of reset and update gate as shown in equation 8- 11.

$$r_t = \sigma(x_t U^r + h_{t-1} W^r) \quad (8)$$
$$z_t = \sigma(x_t U^z + h_{t-1} W^z) \quad (9)$$
$$k = \tanh(x_t U^k + (h_{t-1}.r) W^k) \quad (10)$$
$$h_t = (1 - z).k + z.h_{t-1} \quad (11)$$

Both variants employ the same model architecture. A 5 layered architecture with 2 recurrent layers with 64 and 128 units followed by 2 dense layers having 256 and 512 units and a single unit output layer has been used. All units use the 'linear' activation function, and have a dropout probability of 0.3. The dense layers have been initialised by 'he_normal' initialisation function. The model is trained for 125 epochs using the 'Adam' optimiser as the optimisation algorithm and root mean squared error was used as the loss function.

Stock prices for the immediate t+5 days can be determined by the price action occurring in the past trading week. By performance comparisons in Table 3, it can be observed that a 5 day time window of back propagation for the mode is optimal.

Table 3-This experimentation has been performed on an LSTM model with optimal model configurations.

| Look back Period | RMSE | R^2 Score | Return Ratio | Optimism Ratio | Pessimism Ratio |
|---|---|---|---|---|---|
| No look back | 0.000387 | 0.951347 | 3.1390147 | 0.329762 | 0.165476 |
| 3 Days | 0.000546 | 0.931349 | 4.546969 | 0.427718 | 0.173238 |

| Look back Period | RMSE | R^2 Score | Return Ratio | Optimism Ratio | Pessimism Ratio |
|---|---|---|---|---|---|
| 5 Days | 0.000534 | 0.932749 | 4.428151 | 0.408383 | 0.171257 |
| 8 Days | 0.000690 | 0.913146 | 1.675390 | 0.537259 | 0.131010 |
| 11 Days | 0.000595 | 0.925140 | 5.003220 | 0.451146 | 0.154403 |
| 14 Days | 0.000694 | 0.912629 | 2.407061 | 0.529055 | 0.138014 |

### 3.2 Artificial Neural Network

Artificial neural network[2] is an established method for prediction of stock prices. It is a feed forward neural network, which is trained using back propagation i.e. the computed error at the output layer is fed backwards into the network towards the input node. For this study a 4 layered architecture is used with 64, 128 and 256 units in each layer and an output layer. Every layer has a dropout probability of 0.3 with an activation function of 'Rectified Linear Unit'. The model is trained using 1500 epochs with 'Adam' optimisation function, root mean squared error as the loss function and 'he_normal' as the unit initialisation function.

### 3.3 Support Vector Machine

Support vector machines[3] find a hyperplane which maximise margin from the support vectors and divide the data into 2 different segments. The proximity of data points from the hyperplane will decide the regression value of the point. Kernel methods are transformation techniques into a higher dimension, so that the transformed data can be separable by linear hyperplane in the higher dimension. In this study, a sigmoid kernel is used with a degree 2, regularisation constant 'C' of 2, kernel function gamma of 0.7, and a hinge loss of 0.001.

## 4. Implementation

### 4.1 Trading Strategy

The machine learning model will be used for stock price prediction and to find an automated investment strategy. Price predictions will be carried out via regression. For each trading session the model predicts a price for the $t+5^{th}$ day which is higher than the current price a buy signal is generated and one share of that stock is purchased on the current and sold on the $t+5^{th}$ day. The difference between the actual price on the $t+5^{th}$ day and the current price is the profit/loss. When our model predicts a fall in price a sell signal is generated and the stock is shorted.

### 4.2 Evaluation Metrics

Evaluation of the proposed models in this study have been done using 5 major metrics. R2 score and mean squared error has been used to evaluate the regression performance of the model.

The optimism and pessimism ratios are evaluate the number of instances when model predicts a value 1.5% greater than the actual price. These metrics would define the confidence level we would have in our predictions. If the optimism and pessimism value are both low, the model neither predicts higher nor lower than the actual price often, and has a high confidence level. A higher value may mean that most predictions are not close to the actual price and have low certainty.

The effectiveness of the trading strategy outputted by the model can be evaluated by the return ratio. The return ratio is a ratio between the profits earned by following the strategy outlined by the model against the profits earned by a hold and wait strategy. A high return ratio shows that the model is successful in detect the underlying trend in the stock and can accurately predict when the stock value will rise and fall.

$$OR = \frac{OPT}{Length} \quad (12) \qquad PR = \frac{PES}{Length} \quad (13)$$

$$Where,$$

*OPT = Count of records where [ predicted_price > 1.015*actual_price ]* (14)

*PES= Count of records where [ predicted_price < 0.985*actual_price ]* (15)

*Length = Total Number of Records* (16)

## 5. Results & Discussions

As seen in Table 4, the conventional LSTM model has the highest R2 score of 0.9486, the lowest RMS error of 0.000428 and the lowest average value of the optimism and pessimism ratio. An optimism ratio of 0.3 and a pessimism ratio of 0.2 would mean that the LSTM model has 3 and 2 out of 10 predictions which are greater and lower than 1.5% of the actual price.The GRU model has an almost similar R2 score and a RMS error, but a higher optimism ratio so gives us a higher uncertainty in prediction. The ANN model proves to be inferior in comparison to the other models. A higher RMS error and a lower R2 score, high optimism ratio suggests that this model has lower performance and higher volatility as confirmed by its jagged price action in Figure 1. Price action for LSTM and GRU models in Figure 1, is smooth, and has low volatility like that seen in the actual prices. The SVM based model has a high pessimism ratio and a low optimism ratio suggesting that most price predictions are below the actual price, and there are few over estimated predictions. This problem could be solved by scaling the price predictions by a factor so that they become closer to the actual prices.

The GRU model has the highest return ratio followed by the LSTM and then the ANN and SVM models. The GRU and LSTM models successfully identify the future trends in the market and find out the appropriate entry/exit points. High volatility seen in the ANN models correspond to lower return ratio. SVM predictions lag the actual prices and hence, they are bad trend predictors and output a loss making strategy.

The GRU and the LSTM model outperform both the ANN and the SVM model. They are better predictors of trend and exact prices. The GRU model has a better return ratio indicating that it predicts trends better and has a more robust strategy but has lower certainty of prediction than LSTM's. Hence, is sub-optimal to the LSTM model.

Despite optimal results, this model cannot emulate cases of extreme price dips or sharp price spike. These instances may depict a dramatic event in the economy like an election or a global event impacting the economy. The model can be made resistant to this, if the dataset is augmented with economic factors like interest rate, GDP growth.

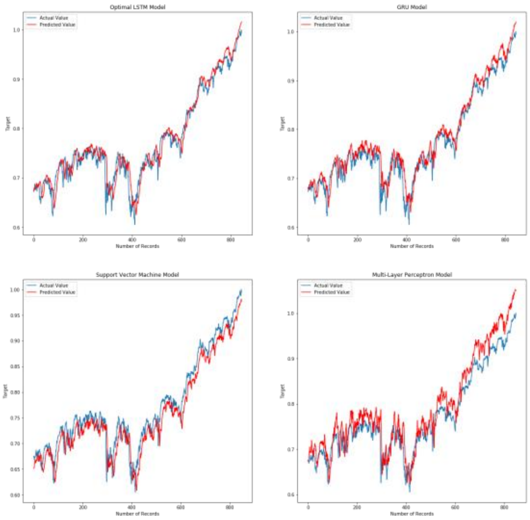


Figure 1- Price Action of Various models.
(Top: Left-LSTM, Right-GRU, Bottom: Left-SVM, Right ANN)

Table 4- Performance Comparison between models

| Model | RMSE | R^2 Score | Return Ratio | Optimism Ratio | Pessimism Ratio |
|---|---|---|---|---|---|
| LSTM | 0.000428 | 0.948616 | 4.308454 | 0.310426 | 0.203791 |
| GRU | 0.000511 | 0.938698 | 5.722242 | 0.447867 | 0.120853 |
| SVM | 0.000543 | 0.934952 | -1.858130 | 0.080094 | 0.631331 |
| MLP | 0.001052 | 0.874004 | 2.478719 | 0.689046 | 0.115430 |

## References

[1] Kwon, Yung-Keun & Moon, Byung-Ro. (2007). A Hybrid Neurogenetic Approach for Stock Forecasting. IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council. 18. 851-64. 10.1109/TNN.2007.891629.
[2]Singh, R. & Srivastava, S. Multimed Tools Appl (2017) 76: 18569

[3]Jigar Patel, Sahil Shah, Priyank Thakkar, K Kotecha, Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques, Expert Systems with Applications, Volume 42, Issue 1, 2015, Pages 259-268
[4]George S. Atsalakis, Kimon P. Valavanis, Surveying stock market forecasting techniques – Part II: Soft computing methods, Expert Systems with Applications, Volume 36, Issue 3, Part 2, 2009, Pages 5932-5941
[5]Jigar Patel, Sahil Shah, Priyank Thakkar, K Kotecha, Predicting stock market index using fusion of machine learning techniques,
Expert Systems with Applications, Volume 42, Issue 4, 2015, Pages 2162-2172
[6]Lamartine Almeida Teixeira, Adriano Lorena Inácio de Oliveira, A method for automatic stock trading combining technical analysis and nearest neighbor classification, Expert Systems with Applications, Volume 37, Issue 10, 2010, Pages 6885-6890
[7]Chen, Kai & Zhou, Yi & Dai, Fangyan. (2015). A LSTM-based method for stock returns prediction: A case study of China stock market. 2823-2824. 10.1109/BigData.2015.7364089.
[8]Erkam Guresen, Gulgun Kayakutlu, Tugrul U. Daim, Using artificial neural network models in stock market index prediction, Expert Systems with Applications, Volume 38, Issue 8, 2011, Pages 10389-10397
[9]Wei Shen, Xiaopen Guo, Chao Wu, Desheng Wu, Forecasting stock indices using radial basis function neural networks optimized by artificial fish swarm algorithm,
Knowledge-Based Systems, Volume 24, Issue 3, 2011, Pages 378-385
[10]Emad W. Saad Danil V. Prokhorov Donald C. Wunsch "Comparative study of stock trend prediction using time delay recurrent and probabilistic neural networks" Neural Networks IEEE Transactions on vol. 9 no. 6 pp. 1456-1470 1998.
[11]Yudong Zhang, Lenan Wu, Stock market prediction of S&P 500 via combination of improved BCO approach and BP neural network, Expert Systems with Applications, Volume 36, Issue 5, 2009, Pages 8849-8854,
[12]Apostolos Nicholas Refenes, Achileas Zapranis, Gavin Francis,
Stock performance modeling using neural networks: A comparative study with regression models, Neural Networks, Volume 7, Issue 2, 1994, Pages 375-388,
[13]-Yahoo Finance, https://in.finance.yahoo.com/quote/SPY?p=SPY, last accessed 2017/10/30.
[14]Yiu-ming Cheung, Lei Xu,Independent component ordering in ICA time series analysis, Neurocomputing, Volume 41, Issues 1–4, 2001, Pages 145-152
[15]Chung, Junyoung & Gulcehre, Caglar & Cho, KyungHyun & Bengio, Y. (2014). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling.
[16]Sak, H & Senior, Andrew & Beaufays, F. (2014). Long short-term memory recurrent neural network architectures for large scale acoustic modeling. Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH. 338-342