

## **On the character of Indian Stock Markets: A Machine Learning Approach**

Saurabh Chauhan

Tushar Jindal

Vaibhav Saini

The Northcap University

### Author's Note

Special thanks to the Northcap University for availing us with the resources and the platform to pursue this line of interest.

## **Abstract**

The enterprise of forecasting the stock market is as old as the market itself, ranging from the many traditional approaches like regression analysis and linear methods like AR, MA, ARIMA and ARMA, and of course fuzzier methods like expert's intuitions and sentiment analysis of news cycles. But owing to the non-linear and dynamic nature of the markets, these methods have a high error rate that has only been improved upon with the advent of deep neural network architectures in the recent decades, like MLPs, ANNs, RNNs, LSTMs and CNNs, which have proven themselves to be excellent approximators of non-linear functions. To forecast is to look within a data to study its inner dynamics and use it for predictive analysis, and it's with deep learning that we've tried to probe into the character of the Indian stock market NSE in particular, and see how well its inner dynamics generalize to other stock markets like the NYSE. To that end we compared two companies from similar industries to check for a possibility of comovement, in this case Reliance listed under the NSE, and AT&T listed under the NYSE. We found that CNNs are the most performant at adjusting to real-time non-linearities while other architectures get biased predictions through over-reliance on past data.

*Keywords:* AR - Auto-regression

MA - Moving Average

ARIMA - Auto-regressive integrated moving average

ARMA - Auto-regressive moving average

MLP - Multi-layered Perceptron

ANN - Artificial neural network

RNN - Recurrent neural network

LSTM - Long Short Term Memory

GRU- Gated Recurrent Units

CNN - Convolutional Neural Networks

NSE - National Stock Exchange

NYSE - New York Stock Exchange

## **On the character of Indian Stock Markets: A Machine Learning Approach**

Neural networks is a collection of deep learning algorithms that attempt to approximate underlying relationships in a set of data through a layered-architecture that was inspired by biological neural architectures. Perhaps the most performant data mining techniques that have been employed by computer scientists in various disciplines for the past few decades, they have gradually been gaining traction in the financial sector as well. Predictive analysis of stock market data has got a seminal role in the world's economy, an active area of research ever since stock markets were found, albeit the conventional forecasting methods were never found to be reliable. With the advances in machine learning in the recent decades, there has been some promise of forecasting the non-linear, dynamical nature of the market. The algorithms used for stock market forecasting can be largely classified as linear (AR, MA, ARIMA, ARMA) and non-linear methods (ARCH, GARCH, Neural Network). In this paper, we will be implementing and assessing four types of neural network architectures, namely Multilayer Perceptron (MLP), Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN) for approximating the closing price of a company based on historical time series data. After training our model on data from companies listed under the National Stock Exchange, we used transfer learning to forecast the closing prices of other companies from the same industry, as well as AT&T, which is listed under the New York Stock exchange, to check for comovement between international markets. Our lines of inquiry are:

1. Which model performs ideally and under what conditions and optimizations?
2. Whether the network was able to generalize its learning to NYSE despite the fact that it was trained with NSE data. This is to investigate any common elements between the two markets.

## **Literature Survey**

We have done a comparative study of many different stock market forecasting techniques, both conventional and deep learning based, across a variety of datasets with different metrics like the S&P 500 index, the NASDAQ exchange rate [1], closing prices of the National Stock Exchange as well as the New York Stock Exchange, and the causal relationship between volume and price.

It is abundantly clear that the stock market is a very dynamical system which linear models fail to explain, and making predictions on previous lag leads to unreliable forecasting. The best models are based on non-linear approaches like ANN, with the highest performance being achieved through sliding window CNN, which most closely addresses the time-critical nature of this problem for investors. The more real-time the predictions, the more useful they are to the investors. CNN that can predict on a short sliding window, seems to perform the best.

These approaches may also be labelled as model independent, since they don't involve fitting the data to pre-existing models, but rather finding the internal relationships between the features. In a study done in 2017 [7] to predict stock prices in TCS, Cipla and Infosys, sliding window CNN outperformed RNN and LSTM [3].

Another study done in 2011 [8] to establish the superiority of machine learning models to conventional approaches, employed Artificial Neural Networks. The basic idea is that stock markets are a chaotic system and the best way to predict stock prices is to use a network's brute force approach to iteratively optimize itself to learn the correct outputs. In that way, it can

approximate any function. Networks are also very good at getting rid of noise, as values that don't contribute to the result get their weights lowered. In conclusion, with enough data and iterations, ANNs outperform all conventional approaches.

To compound the problem, most conventional approaches that financial analysts use are ultimately subjective with varying choices of what features to weigh, filtered by their own worldviews and cognitive biases, while ANNs and CNNs directly tackle the problem by trying to approximate a function for the stock indices within a short window.

Given how time sensitive the market is, one would expect that Recurrent Neural Networks, optimized for learning spatio-temporal data, would perform better than ANNs. But Jovina Roman and Akhtar Jameel [9] found that there wasn't a significant enough improvement. The RNN, with its feedback mechanism, was expected to have a greater prediction accuracy than standard backpropagation, but the results didn't indicate any statistically significant difference. A simple explanation could be that one week's sequence was probably not enough data to train and validate on. In order to account for this lack of information, a second group of experiments instead did the training on a 2 year time window, and tested on the subsequent year. The difference in prediction accuracy was around 10.4%, with a year's worth of training data.

We also wanted to test the efficient market hypothesis (EMH) or theory which states that share prices are supposed to reflect all the relevant information about the market, using the Toda-Yamamoto method [2]. The cited paper checked for a Granger causality relationship

between stock prices and trading volume among Nifty 50 [2] companies, from July 2014 to June 2015. The Toda-Yamamoto [2] methodology was used to test for causality, as data wasn't integrated in the same order. The results could be summarised as: 29 out of 50 companies had a bidirectional causality between price and volume; 15 companies had a unidirectional causal relationship where price caused volume but volume did not cause price and the remaining 6 companies had no causal relationship at all. The study overall showed that the Efficient Market Hypothesis stands on weak grounding.

Another study[5], in an effort to predict the NASDAQ index, used stock exchange rates of NASDAQ to try and fit a robust model. 70 days were selected as the training dataset and 29 days were used for testing the model's predictive power with respect to the NASDAQ index. For this dataset, some optimizations that they explored were the OSS training method and the TANGSIG transfer function.

Another algorithmic optimization[4] we explored was that of Differential Evolution(DE), which is a population-based, stochastic function optimizer. It's actually a more greedy and less stochastic approach than classical evolutionary algorithms. DE combines basic arithmetic operators with the classical evolutionary operators of recombination, mutation, and selection to evolve a randomly generated initial population to a final population. In conclusion, we found that DE-enhanced networks (recurrent computationally efficient functional link neural network(RCEFLANN)) [9] are good for minimizing variance and can be used to forecast the

volatility of stock indices, and henceforth make better informed trading decisions. Its sensitivity to volatility could also be useful for risk management.

The dynamical nature of stocks is also owed to how subjective the notion of value is. Factors like reputation, financial momentum, PR and trust all play a role in how an investor weighs a stock. This has in turn motivated many attempts to try to model the relationship between stock prices and some subjective choice of sentiment metric.

One such study[6] tried to parse out different emotions through natural language processing of real world events as reported in news texts, and how they may affect a stock's value. The events are embedded into a vector, and trained using new neural tensor networks. Then, a CNN was used to model both short-term and long-term predictions on "event embeddings"(news events encoded into input tensors), and how these events affect stock movement. The study found a 6% improvement in S&P index prediction using this Natural Language Processing-based approach.

### **Artificial Neural Networks:**



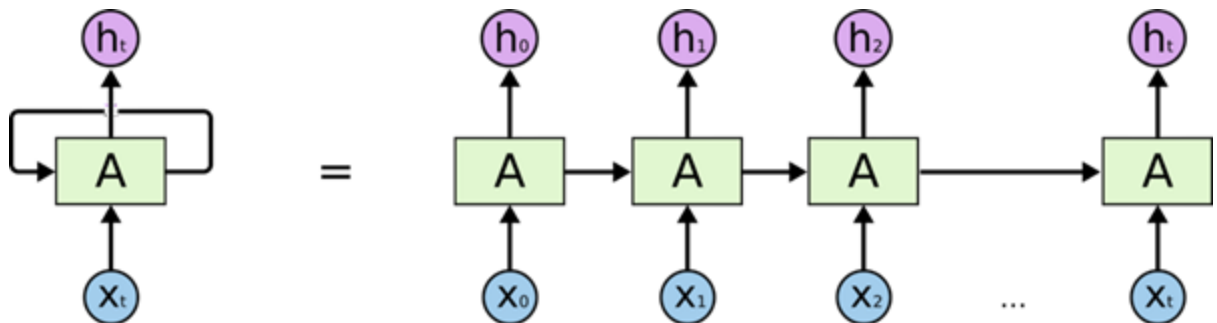
Artificial neural networks, or simply neural networks, are computational methods loosely inspired by biological cognition, in structure and not necessarily in function. The analogy holds in so far as: neurons in our brains are known to fire above a threshold of voltage; artificial neurons, or perceptrons, also produce an output that has to be above a threshold depending on the activation function. Biological neurons have synapses, or connections, that get weakened or strengthened depending on the saliency and frequency of use of that pathway; perceptrons can be arranged in directed, weighted graphs with several layers, and each connection's weight can be dynamically updated according to several architectures and algorithmic techniques. Gradient descent in the back-propagation step of neural nets, is one popular way that the architecture learns to update its weights in direction of minimizing the total loss in the predictions. [10]

### **Multi-layered Perceptrons(MLPs):**

Multi-layered perceptron is an ambiguous term that can refer to a whole class of feed-forward artificial neural networks. It consists of a directed, weighted graph of connected perceptrons(individual units of computation) across input, hidden and output layers. Generally a perceptron “activates” according to a non-linear transformation such as the sigmoid function, which essentially maps the *linear* input features to a *non-linear* probability space for making classifications. [10]

### Recurrent neural networks(RNN):

RNNs are a derivative class of feed-forward neural networks that can use the directed graphs to learn temporal sequences of data, having an internal state or memory to store the inputs, outputs and parameters. There are several different architectures in this class, which differ in the level of connectivity between the nodes, how much “past” data the neuron relies upon, et cetera. Because the network’s reliance on data from past time stamps can sometimes lead to a problem of vanishing gradients as errors get smaller, there are several variants that attempt to solve this. One is Independent RNNs, which only rely on a given neuron’s *own* past states, or LSTM(Long short term memory) which modifies the context of a neuron to a more practical window, avoiding the problem of vanishing gradients somewhat. [10]

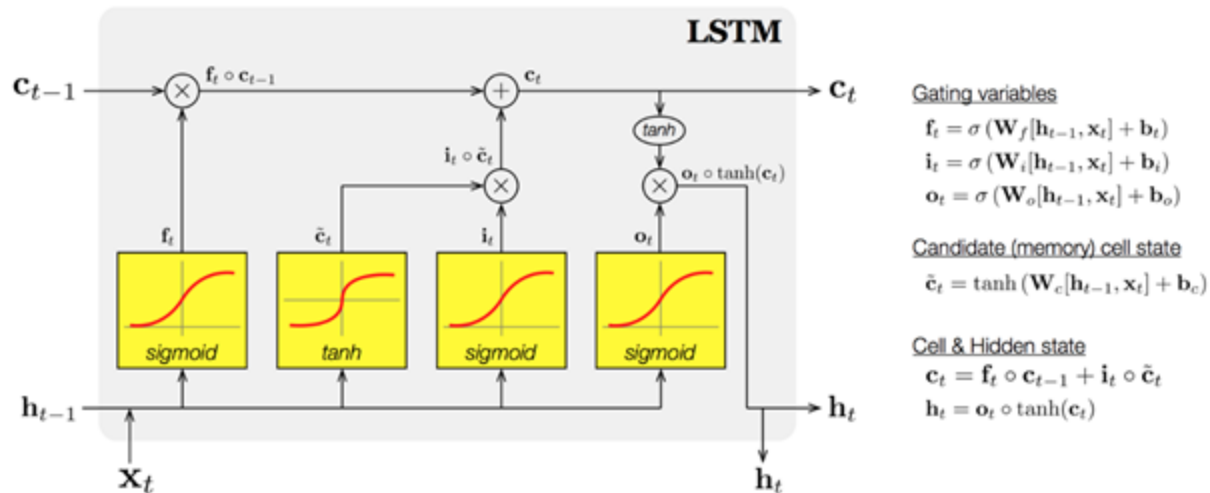


Source: <https://link.springer.com/article/10.1007/s00542-019-04454-8/figures/1>

### Long Short-Term Memory(LSTM):

LSTM is an improvement over RNNs in learning temporal data, as they try to reduce the traditional RNN’s over-dependence on past data, thus reducing the problem of vanishing error gradients during the back-propagation step of training the model.

A unit of LSTM is typically a cell that stores the values of input, output and the *forget* gates. It can be seen that the gated structure of a cell allows for a greater degree of control over the flow of temporal information within the network. [10]



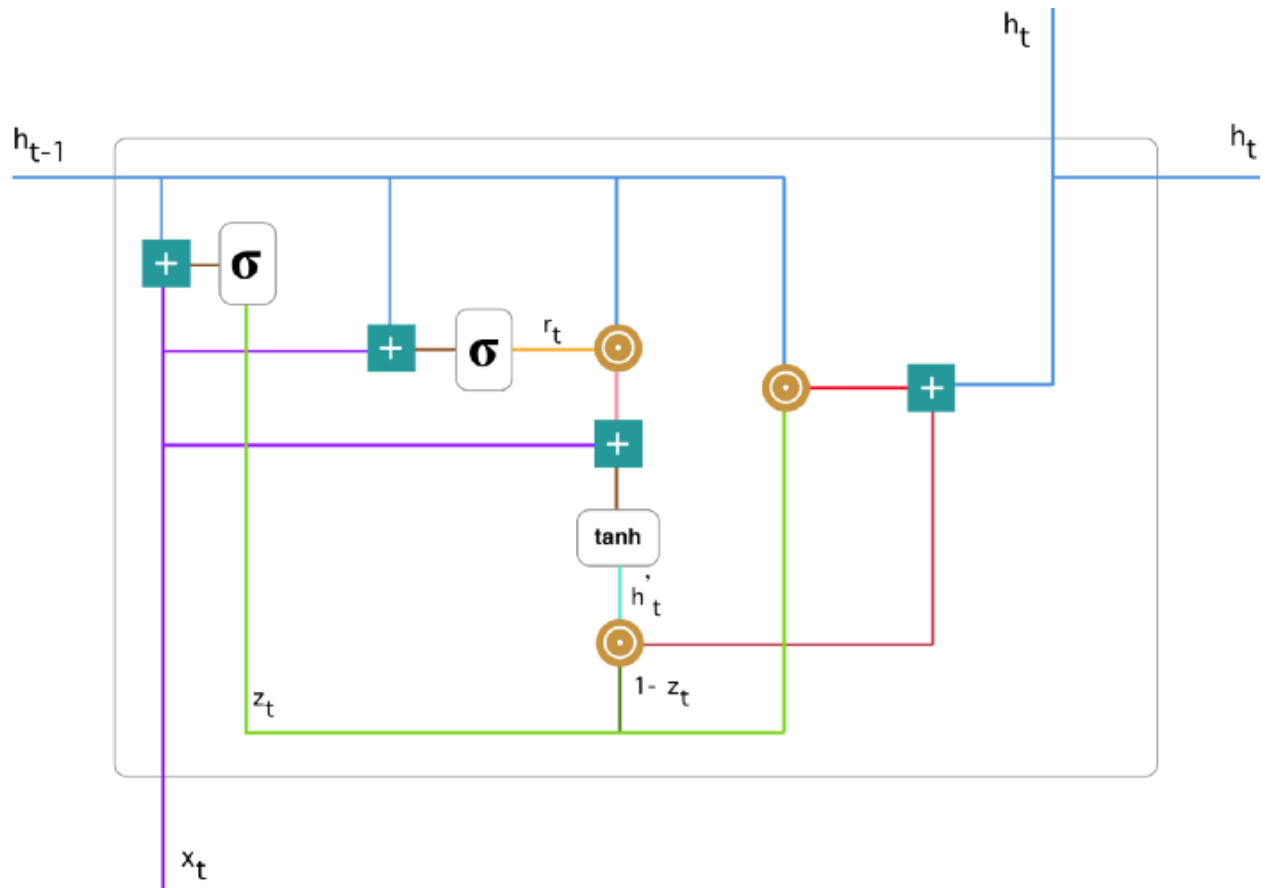
Source:

<https://medium.datadriveninvestor.com/in-artificial-intelligence-new-types-of-networks-are-constantly-generated-by-scientists-a032cd7b77d9>

### Gated Recurrent Units (GRU):

GRU units are yet another way to solve the problem of vanishing gradients, and quite similar to LSTM cells. GRUs have fewer parameters and thus may train a bit faster or need less data to generalize. But, depending on the scale of the data, one may encounter trade-offs with LSTM's expressiveness.

They are a mutation of LSTM cells in that they don't have a forget gate and instead have: reset gate and update gate. Reset gate controls how new input is compiled with previous memory; update gate dictates how much of the previous state to store. Update gate in GRU cells is what input and forget gate are in LSTM cells. [10]



### Convolutional Neural Networks(CNN):

CNNs are a class of deep neural architectures that have gained popularity in the past decade for computer vision and related classification and recognition tasks. But depending on the task's

implementation details, they may also have applications in natural language processing, and more importantly, financial time series data.

One of many architectures inspired from biological systems, CNNs differ from ordinary MLPs in that they're not fully connected graphs, and hence are not equally prone to overfitting.

While the overfitting problem is fixed in other architectures through regularization, CNNs accomplish the same thing through recognizing hierarchies within data, and building on that complexity by layering simpler filters.

Out of all the other architectures, feeding data to CNNs requires the least amount of preprocessing. Its architecture makes it capable of optimizing for the right filters, therefore solving the problem of feature extraction. [10]

### **Methodology:**

After much survey of the available literature, we settled on a few algorithms to test on the Indian Stock Market and see if we could extract commonalities that are universal to foreign stock markets as well. Out of the possible correlations we could look into, which affects how we feed the data into the network, one possible causality test has already been done. That is between stock value and volume, put to the test using the Toda Yamamoto methodology. Most studies done on this have been somewhat inconclusive, and based on the Granger causality test, seemed like a dead end to investigate. Further survey of the stock market tells us that correlating volume with prices is not a good idea as stock can often be undervalued and its value is often very subjective and fuzzy, thus volatile.

We could also use any of these time-series datasets to model stock indices instead, and that'd be a separate line of inquiry altogether, but since we're of the belief that macro-variables tend to not carry enough meaning on a large enough scale for non-linear systems, we instead chose a more approachable target like closing prices based on a sliding window. Stock indices are an aggregate measure, and don't offer much "resolution" in studying the nature of markets or our main question of comovement between the NSE and NYSE, keeping the industry same.

Speaking of resolution, while it's true that CNNs have performed the best in terms of reducing the error rate in shorter windows of time, essentially optimizing for the right amount of past dependence, it should be noted that LS-SVRs(Least Square Support Vector Regressors) have in some cases shown better performance on long term predictions. But here we mostly look into deep learning based algorithms, and how their architectures are optimal for extracting the finer patterns in non-linear data.

### **Sliding window CNNs:**

Sliding window is a much more general algorithm, in which a "window" or a snippet of data is captured, operated upon, then the window slides forward, as the newer portion of the data is operated upon, and so on. Here we implement the same strategy but with CNNs. It is clear that this strategy reduces the problem of vanishing error gradients and overfitting. A sliding window

CNN is essentially better at adapting to sudden fluctuations in the market because it uses that recent time window which other models don't.

### Results

Architecture:	Root Mean Squared Error:
Multiple linear regression	12.2242870345604
Multi-layered Perceptron	13.204090721575254
Recurrent Neural network	42.25192160459433
Long Short Term Memory	45.47787675616713
Gated Recurrent Unit	78.88104529857085
Sliding window CNN	Reliance: 0.020540780226015055

After having decided that sliding window CNN was the most performant out of all the models, linear or non-linear, we decided to use transfer learning to fit this pre-trained network(on Reliance) on other companies as tabulated below:

Sliding Window CNN	Root Mean Squared Error
Wipro	0.016661140253285488
JSW Steel	0.016151207735118064
AT&T	0.024874078156142832

Out of multiple linear regression, MLP, RNN, LSTM and CNN, we found that the best accuracy was yielded by CNN. After having trained this model on Reliance stock, we used those same trained parameters to fit AT&T stock. While we'd need more data to establish any strong correlations in any event, we found that, indeed, stocks from a similar industry often have comovement, or more generally "integration" due to trading, financial and psychological linkage. The model that learned to predict Reliance worked impressively well on AT&T, suggesting that further studies might need to be done to investigate how stock returns are influenced by such abstract and elusive international linkages. A deep learning based approach is just one of many brute forced approaches to extracting meaning out of a large stream of data, and while its inferences are limited, they have their value in so far as they're taken as descriptive. Its predictive value, while impressive, doesn't say a whole lot about the nature of the market, and is largely task-dependent, contingent on industrial linkages, economic phases, and so on.

## **Discussion**

We've seen that despite the impressive predictive abilities of deep neural networks, as powerful function approximators, the actual nature of the stock markets is rather abstract and elusive. Any investigation into its nature reveals that there's no possible mechanistic delineation of what market forces drive what metric according to what definite mathematical law- it's all very non-linear and chaotic. We know from how the market works, that stock returns are largely influenced by an investor's subjective assessment of how well-rated a stock *ought* to be- an area of decision making actively littered by, and filtered through, human fallacies and cognitive biases. Along that vector, some researchers[6] have tried to feed data into natural language



processors to do a sentiment analysis of news cycles, social media content and the like. That certainly seems to be a natural step forward to making stock prediction models more robust, to account for the human element of the market.

## References

- [1] VISUALIZING AND UNDERSTANDING RECURRENT NETWORKS Andrej Karpathy Justin Johnson Li Fei-Fei Department of Computer Science, Stanford University (2015)
- [2] Measuring stock price and trading volume causality among Nifty50 stocks: The Toda Yamamoto method (2016)
- [3] Investigation Into The Effectiveness Of Long Short Term Memory Networks For Stock Price Prediction By Hengjian Jia (2016)
- [4] Rout A. K., Dash P. K., Dash R., and Bisoi R. (2015). "Forecasting financial time series using a low complexity recurrent neural network and evolutionary learning approach." Journal of King Saud University-Computer and Information Sciences 29 (4):536-552
- [5] Moghaddam A. H., Moghaddam M. H., and Esfandyari M. (2016). "Stock market index prediction using artificial neural network." Journal of Economics, Finance and Administrative Science 21
- [6] Ding X., Zhang Y., Liu T. and Duan J. (2015). "Deep learning for event-driven stock prediction." In Ijcai
- [7] Sreelekshmy Selvin, Vinayakumar R, Gopalakrishnan E.A, Vijay Krishna Menon, Soman K.P (2017). "Stock Price Prediction Using LSTM, RNN AND CNN-Sliding Window Model"
- [8] Zabir Haider Khan, Tasnim Sharmin Alin, Md. Akter Hussain (2011). "Price Prediction of Share Market using Artificial Neural Network (ANN)"
- [9] Jovina Roman and Akhtar Jameel (2008). "Backpropagation and Recurrent Neural Networks in Financial Analysis of Multiple Stock Market Returns"
- [10] Deep Learning with Python by François Chollet.

