S&P 500 STOCK PRICE MODELING USING RNN (LSTM)

Prepared by

Abu Horaira Banna

as part of PH582 project

INTRODUCTION

ML approaches have gained a considerable uptake for time-series forecasting in last few years. One of the primary approaches was the artificial neural network (ANN) model [1]. ANNs have been developed in order to mimic the intelligence of the human brain automatically. More specifically, ANNs try to recognize patterns in the input data attempting to provide generalized results based on their known previous knowledge. As confirmed by reported results, ANNs have largely applied in the financial domain. Their main advantage consists in not providing any a priori assumption about the data. In fact, the model is suitable to fit a given time-series by analyzing the features extracted from complex data, as opposed to traditional models, such as ARIMA.

Recently, a considerable amount of literature has investigated the use of RNN and its variants, such as LSTM, for time-series forecasting. Compared to the classical ANN model, these models achieved better results in forecasting problems due to their powerful ability to capture hidden relationships within data. More specifically, LSTM architecture has been designed to learn long-term dependencies. LSTM is able to manage the input/output data flow through its fundamental unit: the cell. A LSTM cell is composed by three different "gates", called input, forget and output gate, which establish to store or discard data by using sigmoid function, called “activation function”. Also, the input and status of the cell is updated through applying the “tanh function”. Advances in natural language processing (NLP) and DL fields have brought the development of sentiment analysis approaches to transform upcoming web news, tweets and financial reports into relevant features. After extracting useful information from data in textual format, they are processed as input time series to perform forecasting.

Tables 1 summarizes several works based on ML/DL models. The authors chose stock prices of National Stock Exchange (NSE) India data to evaluate the proposed pipeline. More specifically, they collected Infosys data for the period from July 2014 to October 2014 as training data and they used stock price for Infosys, Tata Consultancy Service (TCS) and Chemical Industrial and Pharmaceutical Laboratories (CIPLA) from October 2014 to November 2014. The reported results in Table 1 are related to Error Percentage. In particular, they compared results achieved by RNN, LSTM and CNN to ARIMA model. They observed that the ARIMA model present a higher error percentage value than other models, confirming that neural models are more suitable for the prediction of stock markets affected by high volatility [2].

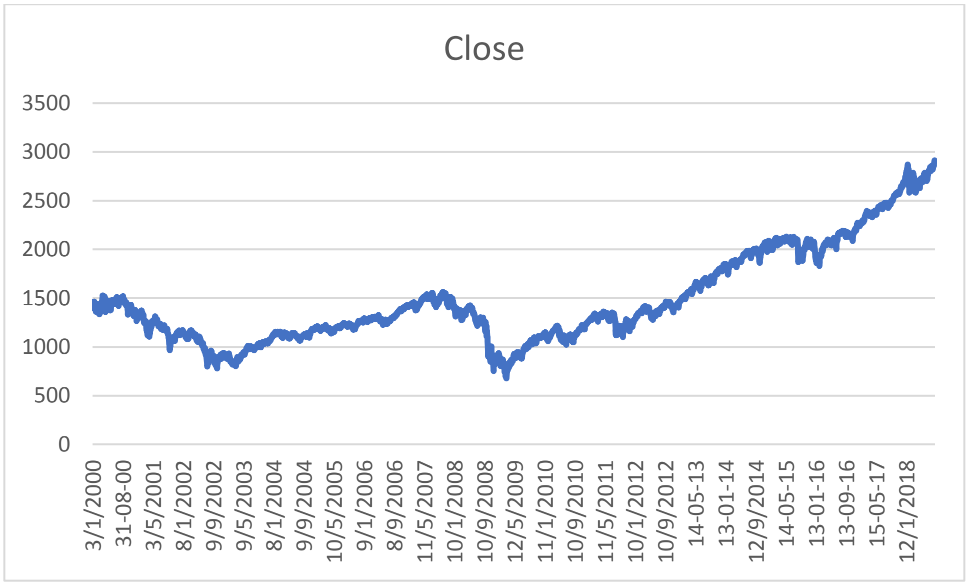
Table 1. Summary of studies based on ML/DL models

Error Percentage		
RNN	3.96	NSE (Infosys)
	7.65	NSE (TCS)
	3.83	NSE (Cipla)
LSTM	4.18	NSE (Infosys)
	7.82	NSE (TCS)
	3.94	NSE (Cipla)
CNN	2.36	NSE (Infosys)
	8.96	NSE (TCS)
	3.63	NSE (Cipla)
ARIMA	31.91	NSE (Infosys)
	21.16	NSE (TCS)
	36.53	NSE (Cipla)

In this work a long-short term memory networks is applied to predict SP 500 future closing prices sequences from 2000 to 2018.

DATA PREPARATION

The data for this work is taken from Ref. [3]. The data set includes open, close, low, high and volume index prices of SP 500 from 2000 to 2018. In figure 1 closing price and volume is shown throughout this time.



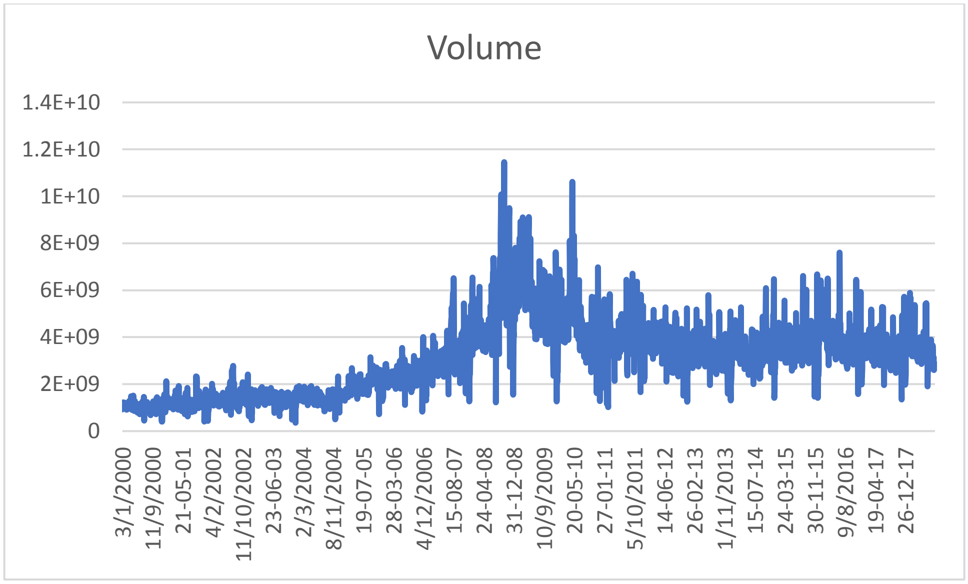


Figure 1. SP 500 closing price and volume from 200 to 2018.

The stock prices is a time series of length N, defined as $S = \{P_0, P_1, \ldots., P_{N-1}\}$. P_i is the index price with $0 < i < N$, N is the number of events in the time series. Each n-sized window of training/testing data is sorted and normalize each one to reflect percentage changes from the start of the window (so the data at point $i = 0$ will always be 0).

$$W_0 = (P_0, P_1, \ldots, P_{(w-1)})$$

$$W_1 = (P_w, P_{w+1}, \ldots, P_{(2w-1)})$$

………..

$$W_t = (P_{tw}, P_{tw+1}, \ldots, P_{(t+1)w-1})$$

The following equations are used to normalize and subsequently de-normalize at the end of the prediction process to get a real-world number out of the prediction.

$$n_i = \frac{P_i}{P_0} - 1$$

$$P_i = P_0(n_i + 1)$$

85% of the data set is used for training the model with remaining for testing.

MODEL DESCRIPTION

The LSTM model in this work has three stacked LSTM layer(s) and each layer contains 100 neurons. Dropout layers are applied to the output of every LSTM cell. The goal of dropout is to remove the potential strong dependency on a dimension so as to prevent overfitting. The training requires two epochs in total; an epoch is a single full pass of all the training data points. In one epoch, the training data points are split into mini-batches of size 32. We send one mini-batch to the model for one BPTT learning. The schematic of the model is show in figure 2.

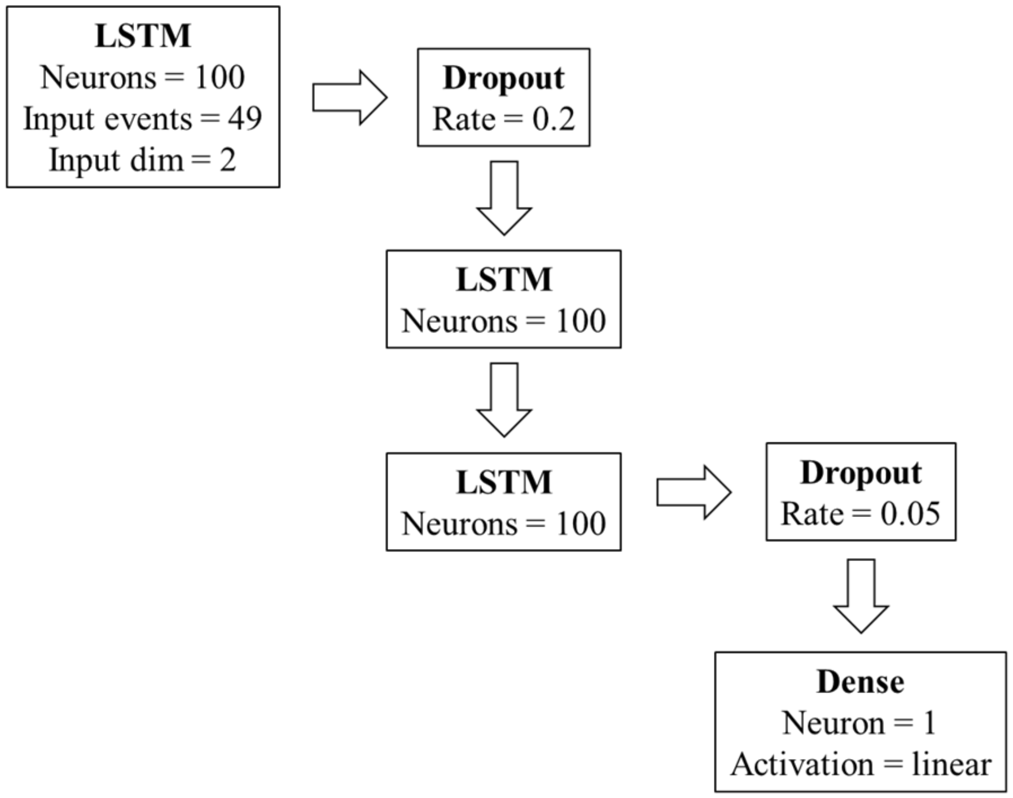


Figure 2. schematic description of the LSTM model for SP 500 price sequence prediction

In the current implementation of the model we use two features to train the model. These features are ‘closing’ and ‘volume’ index of the SP 500 stock data.

RESULTS

Running the data on a single point-by-point prediction as gives something that matches the returns pretty closely. But this is slightly deceptive. Upon a closer examination, the prediction line is made up of singular prediction points that have had the whole prior true history window behind them. Because of that, the network doesn't need to know much about the time series itself other than that each next point most likely won't be too far from the last point. So even if it gets the prediction for the point wrong, the next prediction will then factor in the true history and disregard the incorrect prediction, yet again allowing for an error to be made. Whilst this might not initially sound promising for exact forecasts of the next price point, it does have some important uses. Whilst it doesn't know what the exact next price will be, it does give a very accurate representation of the range that the next price should be in.

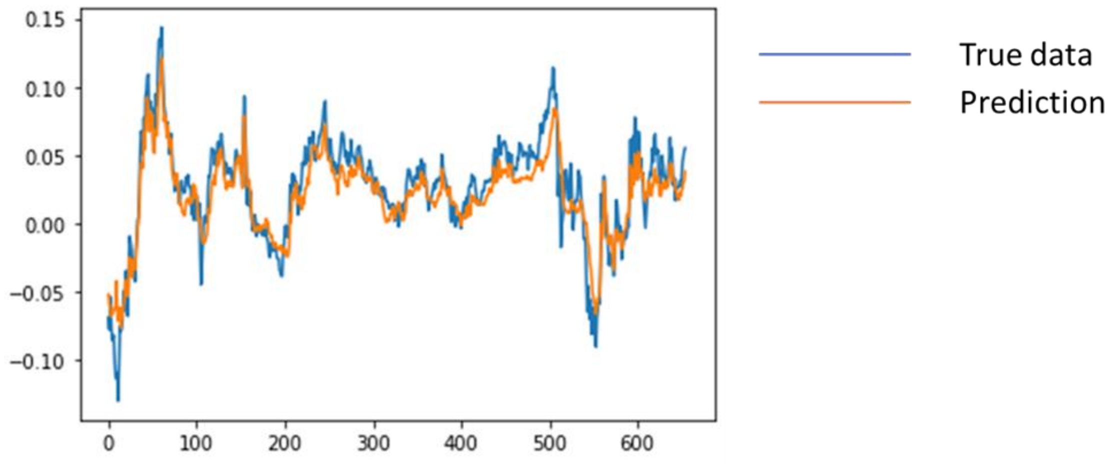


Figure 3. SP 500 point-by-point prediction using LSTM model

This information can be used in applications like volatility forecasting (being able to predict a period of high or low volatility in the market can be extremely advantageous for a particular trading strategy) or moving away from trading this could also be used as a good indicator for anomaly detection. Anomaly detection could be achieved by predicting the next point, then comparing it to the true data when it comes in, and if the true data value is significantly different to the predicted point an anomaly flag could be raised for that data point.

A multi-sequence prediction a blend of the full sequence prediction in the sense that it still initializes the testing window with test data, predicts the next point over that and makes a new window with the next point. However, once it reaches a point where the input window is made up fully of past predictions it stops, shifts forward one full window length, resets the window with the true test data, and starts the process again. In essence this gives multiple trend-line like predictions over the test data to be able to analyze how well the model can pick up future momentum trends.

We can see from the multi-sequence predictions that the network does appear to be correctly predicting the trends (and amplitude of trends) for a good majority of the time series. Whilst not perfect, it does give an indication of the usefulness of LSTM deep neural networks in sequential and time series problems. Greater accuracy could most certainly be achieved with careful hyperparameter tuning.

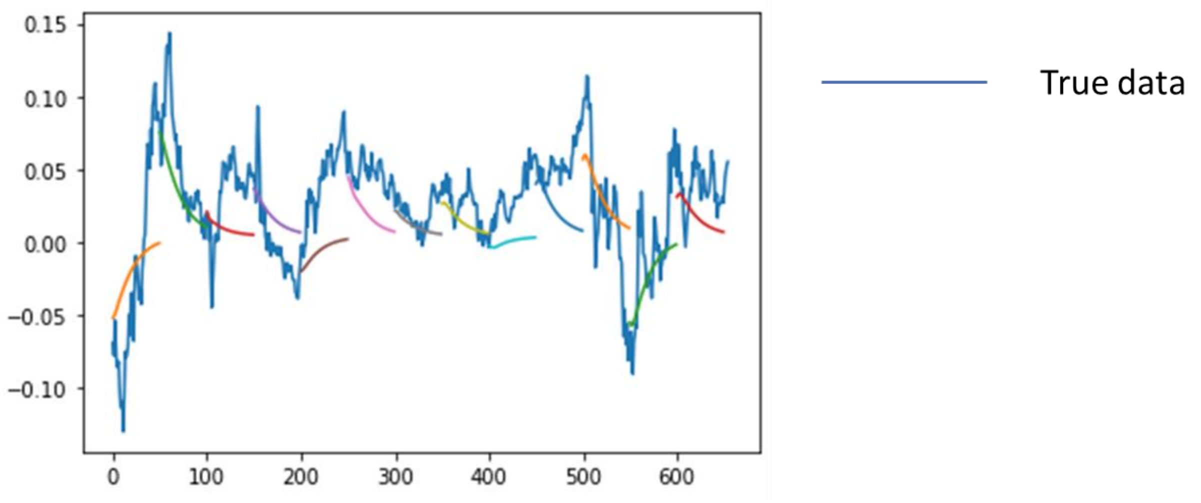


Figure 4. SP 500 multi-dimensional multi-sequence prediction using "Close" & "Volume" index with LSTM model

CONCLUSIONS

LSTMs have been successfully used in a multitude of real-world problems from classical time series issues as described here, to text auto-correct, anomaly detection and fraud detection, to having a core in self-driving car technologies being developed. There are limitations with using the LSTMs described above, specifically in the use of a financial time series, the series itself has non-stationary properties which is very hard to model (although advancements have been made in using Bayesian Deep Neural Network methods for tackling non-stationarity of time series). Also, for some applications it has also been found that newer advancements in attention-based mechanisms for neural networks have out-performed LSTMs (and LSTMs coupled with these attention based mechanisms have outperformed either on their own). As of now however, LSTMs provide significant advancements on more classical statistical time series approaches in being able to model the relationships non-linearly and being able to process data with multiple dimensions in a non-linear fashion.

References

[1] Francesco et al., *Machine Learning for Quantitative Finance Applications: A Survey,* App. Sci. (2019).

[2] Selvin et al., *Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI),* Udupi, India, pp. 1643–1647 (2017).

[3] Jakob Aungiers, ALTUM intelligence

Appendix

Github link of the source code.