Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 175 (2020) 603–608

Procedia Computer Science

www.elsevier.com/locate/procedia

International Workshop on Artificial Intelligence & Internet of Things (A2IoT)
August 9-12, 2020, Leuven, Belgium

Stock Market prediction on High frequency data using Long-Short Term Memory

Zineb Lanbouri**, Said Achchab

National School for Computer Science and Systems analysis, Mohamed V University, Rabat, Morocco

Abstract

High Frequency Trading (HFT) is part of algorithmic trading, and one of the biggest changes that happened in the last 15 years. HFT or nanotrading represents the ability, for a trader, to take orders within very short delays. This paper presents a model based on technical indicators with Long Short Term Memory in order to forecast the price of a stock one-minute, five-minutes and ten-minutes ahead. First, we get the S&P500 intraday trading data from Kaggle, then we calculate technical indicators and finally, we train the regression Long-Short Term Memory model. Based on the price history, alongside technical analysis indicators and strategies, this model is executed, and the results are analyzed based on performance metrics and profitability. Experiment results show that the proposed method is effective as well as suitable for prediction a few minutes before.

Keywords: High Frequency trading; Long-Short Term Memory; Forecasting; Technical Analysis

1. Introduction

Thanks to the rapid development of computing, training data can now be obtained in a very regular way. Traders deal with minute-based, or sometimes even nanosecond-based data. Therefore, it is particularly important to determine how to analyze useful information and decide if a technical approach can be relevant. Predicting and Forecasting the closing price a few minutes ahead is exactly the problem that the LSTM model is trying to solve. By definition, high Frequency Trading, nanotrading or intraday trading can be defined as a complex type of algorithmic trading that relies on low latencies. According to the Security and Exchange Commission [1][2], HFT is characterized by a very high number of orders, proprietary trading and a very short holding period. Therefore, nanotrading presents the following challenges:

* Corresponding author. Tel.: +212-668-78-38-60
E-mail address: zineb.lanbouri@gmail.com

10.1016/j.procs.2020.07.087

- Speed while taking and cancelling orders;
- Difficulty to predict through the noise of volatile markets; ;
- Profit generation from small accumulated margins.

In our ongoing work, we are considering Deep Learning to tackle these challenges and propose a model with the objective of Forecasting the Closing price one to ten-minutes ahead. Forecasting and Times Series predictions have been the object of theoretical and empirical studies for many decades. The Stock Exchange, in particular, tends to be directly influenced by any improvement in that domain. Therefore, the number of variables and the amount of information are very significant in a way that increases profit and decision time as well. Such criteria are the topic of this paper. Forecasting is the prediction of some future event by analyzing the historical data [3] while Time Series is a chronological sequence of observations for given variables $\underset{n}{X}$ over time $\underset{n}{\overset{(t)}{x}}$ [4](see equation 1). Many researchers have exploited the area of Stock market prediction using Deep Learning in order to improve forecasting and generate profits for the investors. Deep Learning is a part of Machine Learning characterized by its non-linearity and representation learning approach while exploring high levels of abstraction that make algorithms close to human brain. These algorithms achieve positive results in domains such as image recognition, natural language processing, self-driving cars, and a number of other areas. .

Deep Learning for stock market prediction: [5] opts for an autoencoder composed of stacked RBMs to extract features.The study is conducted using daily returns and monthly returns in order to enhance momentum which consists in stocks with high past returns that perform months later. M et al. [6]'s objective is to predict the stock price for two companies in the IT sector and another one in the pharmaceutical sector. The Convolutional Neural Network has showed the best results in comparison with the Recurrent Neural Network and Long-Short Term Memory since it doesn't rely on past data. Fischer and Krauss [7] uses Long Short Term Memory algorithm on S&P 500 stock prices and obtain 0.46% as a result. The training data consists of 250 days prior to the prediction, the objectif is to predict one-day ahead with keras. [8] adopts LSTM in comparison with other Machine Learning algorithms using Open, High, Low, Close, Volume and 175 TA features. This paper shows how LSTM performs in terms of accuracy, average and return per operation. And in this study, the selection of LSTM is justified by its ability to handle sequences whilst distinguishing between recent and early examples [9].

Technical indicators: Most of the automated trading systems adopt Technical Analysis [10]. Unlike Fundamental Analysis that is based on external factors like macroeconomy and financial analysis, TA is attributed to historic data and relies on assumptions, for example, prices are defined by the supply-demand relation and changes in supply and demand cause tendencies to reverse, and can be identified in[11]. [12]'s work includes a combination of momentum, volume, volatility and cycle based indicators with a total of 200 features, then the f-test is performed for feature selection. The study showed that the resulting accuracy was highly dependent on the parameters. [13]uses technical indicators for variance, range and bar information to capture market shocks. The author designs an ensemble model ARIMA-GARH-NN to discover hidden intraday patterns. The algorithm of choice here is Long-Short Term Memory which is a special kind of RNN, introduced by Hochreiter and Schmidhuber in 1997. The main contributions of this work are the following: (1)a forecasting model for high frequency trading using LSTM technique; (2)evaluation of the model by comparing Technical Indicators' impact on Forecasting values. The remainder of this paper is organised as follows: Section II presents the methodology and experiment that we are adopting for this project, High Frequency Trading with LSTM, section III shows the results and discussion, and finally section IV concludes the article.

2. Methodology and experiment

$$X = \{\underset{1}{X}, \underset{2}{X}, .., \underset{n}{X}\} where x \overset{(t)}{=} \{\underset{1}{\overset{(t)}{x}}, \underset{2}{\overset{(t)}{x}}, .., \underset{n}{\overset{(t)}{x}}\} \quad (1)$$

We want to predict the Closing price using High Frequency data by combining classical financial models and Long-Short Term Memory algorithms. The overall structure of our data-driven model includes three major components (see figure 1): Data collection, Data preprocessing and Prediction model.

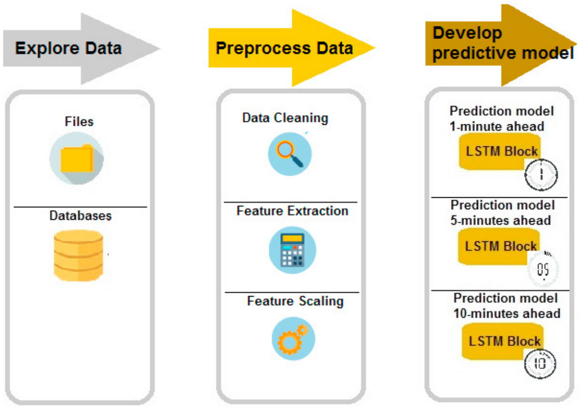


Fig. 1: Workflow

2.1. Data preprocessing

First of all, we collect S&P500 intraday trading data from Kaggle (www.kaggle.com/nickdl/snp-500-intraday-data). The original data files comprise 484 observations. For each observation, every time stamp, Open, High, Low, close price and volume are available from 11/09/2017 to 16/02/2018 with a total of 43148 sequence data. The average spacing between values is 1 minute. Figure 2 plots Amazon stock behaviour, Open, high, Low, Close prices and Volume over one day.

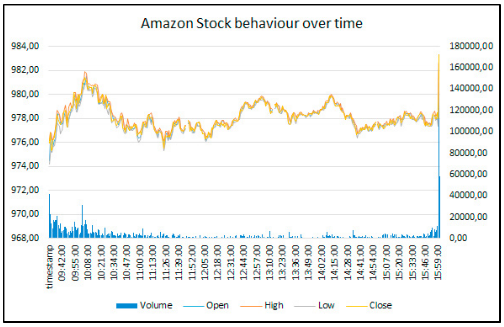


Fig. 2: Amazon Stock Behaviour

Our preprocessing approach follows several steps (see figure 3): step1 for Data cleaning by detecting and removing major errors, step2 for feature creation, consists in calculating some TA metrics, then step3 for feature scaling and normalization.

Data cleaning or cleansing [14] is about detecting invalid and missing values, then handling them. We are considering data that are Missing Completely At Random (MCAR) [15]. The dataset contains missing data that can be split into two categories: values recorded as missing for a a small period and missing values for a long period. Then, we applied two techniques: 1. Value imputation: missing values for a small period (between 1 and 5 instances) are replaced with previous values. 2. Instances discard: missing values for a long period are discarded.

Feature creation Traders are usually confronted with different strategies. We can then distinguish between technical analysis and fundamental analyis. Technical analysis relies only on the stock market, taking into consideration history of the price and some indicators like Moving average and MACD among others. When we take a closer look into works related to stock market forecasting, there are some studies like [16] that believe that ”momentum indicators capture the duration or the turning point of a trend, the MA is often combined with a momentum like MACD”. This work relies, also, on calculated technical indicators EMA (Exponetial Moving Average) - MACD (Moving Average Convergence Divergence) - Bollinger bands and resulting decisions. Exponential Moving Average (EMA) is an indicator of trend which, unlike Simple Moving Average, gives more importance to recent prices.

Moving Average Convergence Divergence (MACD) is a trend-following momentum indicator and when MACD crosses its signal, it can function as a buy and sell signal.

Bollinger bands [17] is a common heuristic used by traders in order to identify turning points in stock prices. Correction of overbought/oversold security signals reflects a change in directional movement.

Feature scaling: The train and test data are standardized where $\tilde{x}$ corresponds to x after standardization and Min is the minimal value of x over time and over the observations, Max is the maximal value of x. After obtaining the predicted output, destandardization is applied in order to get the real profitability value (see equation 2).

$$\tilde{x} = \frac{x - Min}{Max - Min} \tag{2}$$

2.2. LSTM Prediction model

Fig. 3: Preprocessing steps

The objective of the research is to evaluate the performance of the LSTM network on High Frequency Trading. In case of memory and sequential information, the Recurrent Neural Network is a widely used modeling approach by [18]. A recurrent neural network (RNN) can be considered as multiple copies of the same NN, each copy sends the information to its successor using BackPropagation Through Time. RNN is simple but it has a drawback which is known by Curse of dimensionality where gradient can be either vanishing or exploding. LSTM is a solution to that problem, it is suitable because of its ability to have memory and to distingush between recent and older data using gates, see figure 4 below adapted from [19].

2.3. LSTM description

Proposed by [21], the LSTM architecture consists of a set of sub-networks, called memory blocks. Each block contains:

- Memory cell : stores state
- Front door : controls what to learn
- Forget door : controls what to forget
- Exit door : controls the amount of content to modify

The LSTM unit can decide to keep the existing memory via the doors. This memory therefore allows to: . Forget (clear memory) . Input (add to memory) . Output (recover from memory).

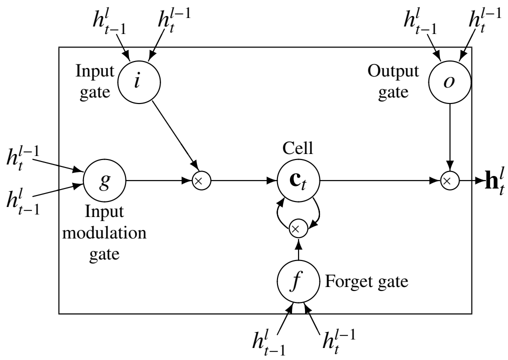


Fig. 4: A graphical representation of LSTM memory cells used in this paper (inspired from [20]).

The LSTM architecture used in our experiments is given by the equations (3) [22][20]: $c_t^l \in \mathbb{R}^n$ is a vector of memory cell, i for the input gate, f for the forget gate, o for the output gate, h for the hidden state and x for the modulation gate. In these equations, sigm and tanh are applied element-wise.

3. Results and discussion

$$\begin{pmatrix} i \\ f \\ o \\ x \end{pmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \tanh \end{pmatrix} T_{2n,4n} \begin{pmatrix} z_t^{l-1} \\ z_{t-1}^{l} \end{pmatrix}$$
$$c_t^l = f \odot c_{t-1}^l + i \odot x$$
$$z_t^l = o \odot \tanh(c_t^l) \tag{3}$$

We split the dataset into the training set from 11/09/2017 9:30 A.M. to 17/01/2018 11:50 A.M., and the validation set from 17/01/2018 11:51A.M. to 16/02/2018 03:59A.M. Then, experiments were carried out to predict the one-minute, five-minutes and ten-minutes ahead price. The key idea is to verify how close each model is to reality in order to know the extent of the risk that the user would take while predicting x-minutes ahead. To further improve our analysis, we apply for every observation a model with and without technical indicators (table 1for RMSE values).

3.1. *Evaluation*

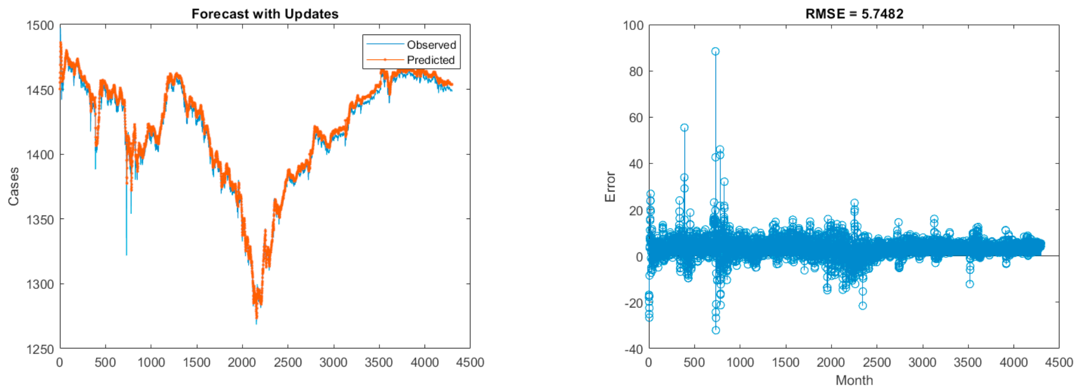
Without Technical indicators
The input layer has a dimensionality of 5 features, that consist of Open, High, Low, Close prices and Volume which is connected to 10 hidden nodes.
With Technical indicators
The number of features is 10: Open, High, Low, Close prices, Volume, EMA12, EMA25, MACD, Bollinger Up and Bollinger Down. In order to evaluate the network, RMSE is calculated for the standardized test part.

Table 1: RMSE values with and without Technical Indicators

	1-minute ahead	5-minutes ahead	10-minutes ahead
without TI	0.0018	0.0046	0.0046
with TI	0.0721	0.0201	0.0108

3.2. *Study Case: Amazon*

We study the effectiveness of our approach in details in this Amazon study case with and without Technical Indicators. The forecasting performance is compared in terms of root mean squared error on the Test part with real values (without standardization) (RMSE, see fig: 5(b) and fig: 6(b)). As shown in fig: 5(a) and fig: 6(a), we can see that we get a better performance without Technical Indicators specially in Amazon's case. RMSE is in average 8.24 (see fig 6(a)) while using Technical Indicators as opposite to 5.74 (see fig 5(a)) without.



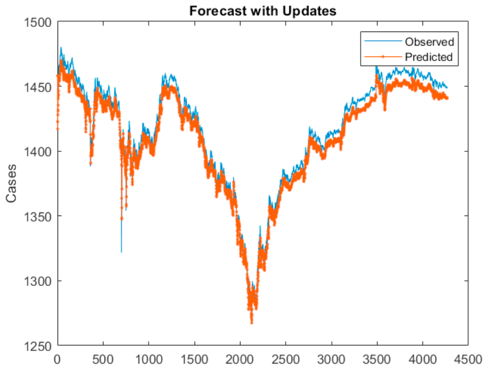
(a) Comparison between Forecast and Observed values

(b) RMSE for Amazon Test part

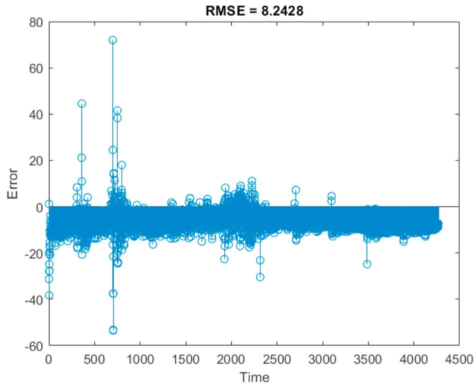
Fig. 5: Without Techncal Indicators

4. Conclusion and future work

Traders and stakeholders rely, nowadays, on many complex parameters and financial model calculations in order to make the right buy-sell-hold decision. The accuracy remains below 50%. That's why this study is about establishing the influence of Technical indicators in forecasting. This model is based on the Long-Short Term Memory algorithm using High Frequency historical data. It confirms that the Closing price can be predicted 10-minutes ahead, 5-minutes ahead and with a better performance one-minute ahead without the use of Technical Indicators. For future work, we attempt to focus on Sampling and Back-testing in order to better master this domain. Furthermore, Deep Learning and LSTM are very promising, we can then expect to improve this model and achieve online prediction for forecasting.



(a) Comparison between Forecast and Observed values using Technical Indicators



(b) RMSE for Amazon Test part with Technical Indicators

Fig. 6: With Technical Indicators

References

[1] C. Lattemann, P. Loos, J. Gomolka, H.-P. Burghof, A. Breuer, P. Gomber, M. Krogmann, J. Nagel, R. Riess, R. Riordan, et al., High Frequency Trading, Business & Information Systems Engineering 4 (2012) 93–108.

[2] A. J. Menkveld, High frequency trading and the new market makers, Journal of financial Markets 16 (2013) 712–740.

[3] S. Selvin, R. Vinayakumar, E. Gopalakrishnan, V. K. Menon, K. Soman, Stock price prediction using lstm, rnn and cnn-sliding window model, in: 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE, 2017, pp. 1643–1647.

[4] P. Malhotra, L. Vig, G. Shroff, P. Agarwal, Long short term memory networks for anomaly detection in time series, in: Proceedings, Presses universitaires de Louvain, 2015, p. 89.

[5] L. Takeuchi, Y.-Y. A. Lee, Applying deep learning to enhance momentum trading strategies in stocks, in: Technical Report, Stanford University, 2013.

[6] H. M, E. A. Gopalakrishnan, V. Menon, S. Kp, Nse stock market prediction using deep-learning models, Procedia Computer Science 132 (2018) 1351–1362. doi:10.1016/j.procs.2018.05.050.

[7] T. Fischer, C. Krauss, Deep learning with long short-term memory networks for financial market predictions, European Journal of Operational Research 270 (2018) 654–669.

[8] D. M. Nelson, A. C. Pereira, R. A. De Oliveira, Stock market's price movement prediction with LSTM neural networks, International Joint Conference on Neural Networks (IJCNN) (2017) 1419–1426.

[9] I. Sutskever, O. Vinyals, Q. V. Le, Sequence to sequence learning with neural networks, in: Advances in neural information processing systems, 2014, pp. 3104–3112.

[10] B. Huang, Y. Huan, L. D. Xu, L. Zheng, Z. Zou, Automated trading systems statistical and machine learning methods and hardware implementation: a survey, Enterprise Information Systems 13 (2019) 132–144.

[11] C. D. Kirkpatrick II, J. A. Dahlquist, Technical analysis: the complete resource for financial market technicians, FT press, 2010.

[12] A. Dingli, K. S. Fournier, Financial time series forecasting-a machine learning approach, Machine Learning and Applications: An International Journal 4 (2017) 3.

[13] J. Sun, K. Xiao, C. Liu, W. Zhou, H. Xiong, Exploiting intra-day patterns for market shock prediction: A machine learning approach, Expert Systems with Applications 127 (2019) 272–281.

[14] E. Rahm, H. H. Do, Data cleaning: Problems and current approaches, IEEE Data Eng. Bull. 23 (2000) 3–13.

[15] M. Saar-Tsechansky, F. Provost, Handling missing values when applying classification models, Journal of machine learning research 8 (2007) 1623–1657.

[16] K.-Y. Kwon, R. J. Kish, Technical trading strategies and return predictability: Nyse, Applied Financial Economics 12 (2002) 639–653.

[17] M. Butler, D. Kazakov, A learning adaptive bollinger band system, in: 2012 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr), IEEE, 2012, pp. 1–8.

[18] C. L. Dunis, X. Huang, Forecasting and trading currency volatility: An application of recurrent neural regression and model combination, Journal of Forecasting 21 (2002) 317–354.

[19] M. Dixon, D. Klabjan, J. H. Bang, Classification-based financial markets prediction using deep neural networks, Algorithmic Finance (2017) 1–11.

[20] W. Zaremba, I. Sutskever, O. Vinyals, Recurrent neural network regularization, arXiv preprint arXiv:1409.2329 (2014).

[21] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (1997) 1735–1780.

[22] A. Graves, N. Jaitly, A.-r. Mohamed, Hybrid speech recognition with deep bidirectional lstm, in: 2013 IEEE workshop on automatic speech recognition and understanding, IEEE, 2013, pp. 273–278.