International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 – 5161

Available at http://inpressco.com/category/ijcet

*Research Article*

# Mid-Price Stock Prediction with Deep Learning

**Mr. Sushilkumar Deshmukh and Prof. S. K. Sonkar**

M.E. Computer Department, Amrutvahini College of Engineering

Received 10 Nov 2020, Accepted 10 Dec 2020, Available online 01 Feb 2021, **Special Issue-8 (Feb 2021)**

***Abstract***

*Mid-price movement prediction based on limit order book data and historical data is a challenging task due to the complexity and dynamics of the limit order book and historical movements of stock data. So far, there have been very limited attempts for extracting relevant features based on limit order book data. In this paper, we address this problem by designing a new set of handcrafted features and performing an extensive experimental evaluation on both liquid stocks. More specifically, we present an extensive set of econometric features that capture the statistical properties of the underlying securities for the task of midprice prediction. The experimental evaluation consists of a head-to-head comparison with other handcrafted features from the literature and with features extracted from a long short-term memory autoencoder by means of a fully automated process. Moreover, we develop a new experimental protocol for online learning that treats the task above as a multi-objective optimization problem and predicts i) the direction of the next price movement and ii) the number of order book events that occur until the change takes place. In order to predict the mid-price movement, features are fed into seven different deep learning models based on multi-layer perceptrons, convolutional neural networks, and long short-term memory neural networks. The performance of the proposed method is then evaluated on liquid stocks. For some stocks, results suggest that the correct choice of a feature set and a model can lead to the successful prediction of how long it takes to have a stock price movement.*

***Keywords:*** *Deep learning, econometrics, stock trading, limit order book, mid-price, NSE stock data*

## Introduction

The automation of financial markets has increased the complexity of information analysis. This complexity can be effectively managed by the use of ordered trading universes like the limit order book (LOB). LOB is a formation that translates the daily unexecuted trading activity in price levels according to the type of orders (i.e., bid and ask side). The daily trading activity is a big data problem since millions of trading events take place inside a trading session. Information extraction and digital signal (i.e., time series) analysis from every trading session provide the machine learning (ML) trader with useful instructions for orders, executions, and cancellations of trades.

We choose econometrics as motivation for our handcrafted features since it is the field of financial engineering that captures the empirical evidence of microstructure noise and the causality of the data. Our data comes with variations in prices, known in the financial literature as volatility a measure that we incorporate into our handcrafted features. Despite the general perception in the academic literature that volatility itself is not a factor that affects stock returns, ample evidence exists to support the opposite.

We perform our analysis based on deep learning models which have recently been proposed for financial time series analysis. These models vary from multi-layer perceptrons (MLP) to convolutional neural networks (CNN) and recurrent neural networks (RNN) like LSTM. For our experiments, we use NSE Stocks datasets from Moneycontorol.

## Literature Review

High-frequency LOB data analysis has captured the interest of the machine learning community. The complex and chaotic behavior of the data in flow gave space to the use of non-linear methods like the ones that we see in the machine and deep learning. For instance, Zhang et al. utilize neural networks for the prediction of Baltic Dry index and provide a head-to-head comparison with econometric models. The author in develops a new type of deep neural network that captures the local behavior of a LOB for spatial distribution modeling. Dixon applies RNN on S&P500 E-mini futures data for a metric prediction like price change forecasting. Minh et al. also propose RNN architecture for short-term stock predictions by utilizing successfully financial news and sentiment dictionary. In, authors apply a combined neural

network model based on CNN and RNN for midprice prediction.

Metrics prediction, like mid-price, can be facilitated by the use of handcrafted features. Handcrafted features reveal hidden information as they are capable of translating time-series signals to meaningful trading instructions for the ML trader. Several authors worked towards this direction. These works present a limited set of features which varies from raw LOB data to change of price densities and imbalance volume metrics. Another work that provides a wider range of features is presented by Ntakaris et al. The authors there extract handcrafted features based on the majority of the technical indicators and develop a new quantitative feature based on logistic regression, which outperformed the suggested feature list.

**Problem Statement**

The problem under consideration is the midprice movement prediction based on highfrequency LOB data. More specifically, we use message and limit order books as input for the suggested features. Message book (MB), contains the flow of information which takes place at every event occurrence. The information displayed by every incoming event includes the timestamp of the order, execution or cancellation, the id of the trade, the price, the volume, the type of the event (i.e., order, execution or cancellation), and the side of the event (i.e., ask or bid).

The main advantage of an order book is that it accepts orders under limits (i.e., limit orders) and market orders. In the former case, the trader/broker is willing to sell or buy a financial instrument under a specific price. In the latter case, the action of buying or selling a stock at the current price takes place. LOBs accept orders by the liquidity providers who submit limit orders and the liquidity takers who submit market orders. These limit orders, which represent the unexecuted trading activity until a market order arrives or cancellation takes place, construct the LOB that is divided into levels. The best level consists of the highest bid and the lowest ask price orders, and their average price defines the so-called mid-price, whose movement we try to predict.

We treat the mid-price movement prediction as a multi-objective optimization problem with two outputs { one is related to classification and the other one to regression. The first part of our objective is to classify whether the mid-price will go up or down and the second part { the regression part is to predict in how many events in the future this movement will happen. To further explain this, let us consider the following example: in order to extract the intraday labels, we measure starting from time tk, in how many events the mid-price will change and in which direction (i.e., up or down). For instance, the mid-price will change in 10 events from now, and will go up. This means that our label at time k is going to be f1,10g, where 1 is the direction of mid-price and 10 is the number of events that need to pass in order to see that movement taking place. IV Proposed Methodology

Our objective is to provide informative handcrafted features to ML traders and market makers for the task of mid-price movement prediction. Prediction of this movement requires in-depth analysis in terms of data selection (e.g., liquid or illiquid stocks) and experimental protocol development. For these reasons, our analysis consists of two NSE Stocks Data based two experimental protocols. The first protocol, named Protocol I, is based on online prediction for every 10-block rolling events, and we introduce it here for the first time. The second protocol, named Protocol II and is based on midprice movement prediction with 10-event lag. Both protocols are event driven, which means that there are no-missing values. However, Protocol II is based on independent 10-block events, which creates a lag of 10 events. Some of the suggested features can partially overcome this problem by finding averages or other types of transformations inside these blocks, but, still some information will be parsed. A possible solution to this problem comes from Protocol I where every single trading event is taken into consideration and, as a result, there are no missing values. We should also mention that LOB data is exposed to bid-ask4 bounce effect which may inject bias. We leave this topic for future research, where we plan to increase the rolling event block size in Protocol I since a wider block will, potentially, improve stability.

*Protocol I*

Both datasets convey asynchronous information varying from events taking place at the same millisecond to events several minutes apart from each other. In order to address this issue, We develop Protocol I, which utilizes all the given events in an online manner. More specifically, our 13 protocol extracts feature representation every ten events with an overlap of nine events for every next feature representation. We decided to use a 10-window block for our experiments due to the frequency 5 of the stationarity present in both datasets. In order to identify whether our time series have unit roots, we perform an Engle-Granger cointegration test6, with focus on the augmented Dickey-Fuller test, on the pair Ask & Bid prices from LOBs level I. The hypothesis test shows that there is a consonant alternation between its states (i.e. 1 for non-stationarity and 0 for stationarity of the suggested time series), which occurs several times during the day. neural networks are capable of identifying underlying processes of a non-stationary time series. Neural networks are nonlinear and nonparametric adaptive-learning filters which operate with fewer assumptions compare to more traditional time series models like ARIMA and GARCH.

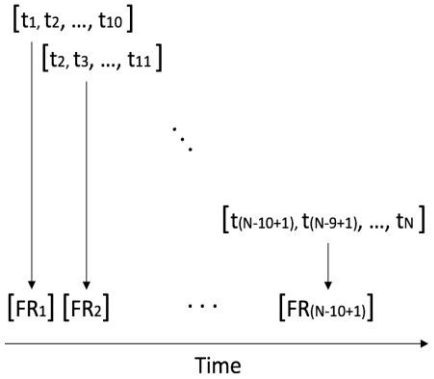
*Protocol II*

Protocol II is based on independent 10-event blocks for the creation of the feature representations as this can be seen in the plot. More specifically, feature representations are based on the information that can

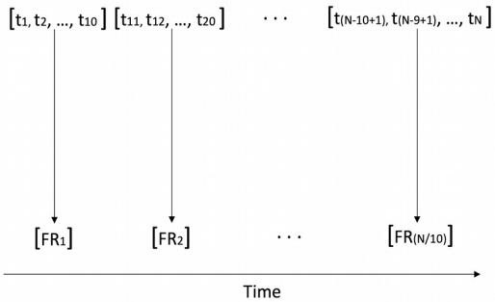
be extracted from 10 events each time with these 10-event blocks independent from each other. Protocol II treats the problem of mid-price movement prediction as a three-class classification problem, with three states: up, down, and stationary condition for the mid-price movement. These changes in the mid-price are defined by means of the following calculations:

$$l_t = \begin{cases} 1, & \text{if} \frac{m_a(t)}{MP(t)} > 1 + \alpha \\ -1, & \text{if} \frac{m_a(t)}{MP(t)} < 1 - \alpha \\ 0, & \text{otherwise} \end{cases}$$

where MP(t) is the mid-price at time t, events with window size r = 10, and  determines the significance of the mid-price movement which is equal to 2 * 10 ^ -5.




Protocol I: Feature extraction in an online manner
with zero lag delay




Protocol II: Feature extraction with 10 events lag

**Results & Discussion**

In this section, we provide results of the experiments we conducted, based on two massive LOB datasets from the NSE stock market (i.e., two stocks: HDFC Bank & Prince Pipes).We also discuss the performance of the handcrafted feature extraction universe for midprice movement prediction and test its efficacy against a fully automated process. What is more, we make a head-to-head comparison of the three handcrafted feature sets, namely: i) "Limit Order Book (LOB):L" ii) "Tech-Quant:T-Q", based on [40], and iii) "Econ:E", which uses econometric features. Finally, we compare these three sets of handcrafted features with features extracted based on an LSTM autoencoder. Latent representations are extracted after training an LSTM AE. This training employs an extensive grid search, in which the best performance is reported. The grid search is based on symmetrical, asymmetrical, shallow, deep, overcomplete, and under complete LSTM AE. The provided options vary from: i) the encoder with maximum depth up to four hidden LSTM layers with different numbers of filters varying according to the list f128, 64, 18, 9g, ii) the decoder with maximum depth up to four hidden LSTM layers with different numbers of filters varying according to the list f128, 64, 18, 9g, and iii) the latent representation with different options varying according to the list f5, 10, 20, 50, and 130g. The best performance reported is based on a symmetrical and under complete LSTM AE of four hidden LSTM layers with 128, 64, 18, and 9 filters respectively, and 10 for the latent representation vector size. The list of the suggested grid different filter options. Further analysis on the topic is required search is limited; however, we believe it provides a wide range of combinations in order to make a fair comparison of a fully automated feature extraction process against advanced handcrafted features. We should also mention that, despite the extensive grid search on the LSTM AE, we limited our search to up to four hidden units for the encoding and decoding parts with four .

Protocol I and Protocol II use three types of deep neural networks as classifiers and regressors. In particular, we utilize five different MLPs, two CNNs, and two LSTMs. Motivation for choosing MLPs is the fact that such a simple neural network can perform extremely well when descriptive handcrafted features are used as input. The next type of neural network that we use is CNN. The first CNN, named "CNN 1", whereas the second one, named "CNN 2" is based on the grid search that we describe below. The last type of neural network that we utilize is LSTM. We use two different architectures: the first one, named "LSTM 1", is based on, and the second one, named "LSTM 2" is based on LSTM with attention mechanism. In total, we train independently nine deep neural networks for each of the two experimental protocols separately.

**Conclusion**

In this paper, we extracted handcrafted features based on the econometric literature for mid-price prediction using deep learning techniques. Our work is the first of its kind since we do not only utilize an extensive feature set list, based on econometrics for the mid-price prediction task, but we also provide a fair comparison with two other existing state-of-the-art handcrafted and fully automated feature sets . Our extensive experimental setup, based on liquid and illiquid stocks (i.e., NSE stocks) showed superiority of the suggested handcrafted feature sets against the fully automated process derived from an LSTM AE. What is more, our research sheds light on the area of deep learning and feature engineering by providing information based on online mid-price predictions. Our findings suggest that extensive analysis of the input signal leads to high forecasting performance even with simpler neural network architects like shallow MLPs, particularly when advanced features capture the

relevant information edge. More specifically, econometric features and deep learning predicted that the mid-price would change direction in a millisecond duration for Amazon and the Joint (i.e., training on both HDFC Bank and Prince Pipes) cases. Although these results are promising, our study here also suggests that selection of features and models should be differentiated for liquid and illiquid stocks.

## References

[1] Alberg, J. & Lipton, Z. C. (2017). Improvingfactor-based quantitative investing by forecasting company fundamentals. arXiv preprint arXiv:1711.04837.

[2] Andersen, T. G. & Bollerslev, T. (1998). Answering the skeptics: Yes, standard volatility models do provide accurate forecasts. International Economic Review, pages 885{905.

[3] Andersen, T. G., Bollerslev, T., & Diebold, F.X. (2010). Parametric and nonparametric volatility measurement. In Handbook of Financial Econometrics, Vol 1, chapter 2, pages 67{137. Elsevier B.V.

[4] Barndorff-Nielsen, O., Kinnebrock, S., &Shephard, N. (2010). Measuring downside risk: Realised semivariance. Volatility and Time Series Econometrics: Essays in honour of Rob Engle.

[5] Barndorff-Nielsen, O. E., Hansen, P. R.,Lunde, A., & Shephard, N. (2008). Designing realized kernels to measure the ex post variation of equity prices in the presence of noise. Econometrica, 76(6):1481{1536.

[6] Barndorff-Nielsen, O. E., Hansen, P. R.,Lunde, A., & Shephard, N. (2009). Realized kernels in practice: Trades and quotes. The Econometrics Journal, 12(3):C1{C32.

[7] Barndorff-Nielsen, O. E. & Shephard, N.(2002). Econometric analysis of realized volatility and its use in estimating stochastic volatility models. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 64(2):253{280.

[8] Barndorff-Nielsen, O. E. & Shephard, N.(2004). Power and bipower variation with stochastic volatility and jumps. Journal of Financial Econometrics, 2(1):1{37.

[9] Barndorff-Nielsen, O. E. & Shephard, N.(2006). Econometrics of testing for jumps in financial economics using bipower variation. Journal of Financial Econometrics, 4(1):1{30. Available from: http://dx.doi.org/10.1093/jjfinec/nbi022. [10] Baydin, A. G., Pearlmutter, B. A., Radul, A. A., & Siskind, J. M. (2015). Automatic differentiation in machine learning: a survey.