Modeling high-frequency limit order book data with LSTM

Chenxi Xiang[1], Weiming Luo[1] and Yiyang Qi[1]

Abstract— **The ability to give precise and fast prediction for the price movement of stocks is the key to profitability in the market. Intuitively, the low level of granularity contains high level of details, which means there could be a large amount of information in the limit order book data. However, due to the complexity and dynamics of the LOB, there have been few attempts for deploying deep learning methods based on millisecond limit order book data. This project examines the application of long-short term memory neural networks to the problem of predicting directional price movements from equities limit order book data and sheds some light on modern quantitative trading based on deep learning. The performance of the proposed method is then evaluate on liquid stock TSLA. Our source code is publicly available at `https://github.com/ProV1denCEX/DL_HFT`.**

I. INTRODUCTION

The price discovery process of financial instruments has caught a lot of attention from many market participants and is a main topic in the market microstructure discipline. Investors are willing to construct better price forecasting models, liquidity providers expect to better understand the market shape and reactions at a very specific moment in order to optimize the bid-ask prices. However, modelling financial market as a high-frequency trading strategy based on limit order book data is not trivial, since the LOB consists of millions of unexecuted trading activities in price levels according to the type of orders every day. Different from long-term investors, high-frequency traders profit from a low margin of the prices changes in a very short time horizon, which requires the capability to observer the dynamics of the market precisely. Meanwhile, the microstructure dataset at the resolution of individual orders, executions, hidden liquidity, and cancellations can be gigantic due to its very fine granularity. For US equities, full order book data is around 1TB/day, even the Level I quotes data is 100GB/day. As for standard methods and theories in finance, they can be ill-equipped to capture highly non-linear interactions in prediction problems and deal with large-scale datasets.

With the development of deep learning, it is possible to gain insights into correlations in markets as complex systems. One of its primary advantage is the computationally and informationally efficient structures for inferring good predictive models from large data sets when compared with those heavy machine learning methods for time series classification and regression problems [1], and thus deep learning is a natural candidate for those problems raised by real trading scenarios based on microstructure data. There have been plentiful researches on stock price prediction with deep learning based on daily datasets. For instance, Masaya Abe and Kei Nakagawa [2] from Nomura Asset Management present a cross-sectional daily stock price prediction framework using deep learning for actual investment management. Ben Moews and Gbenga Ibikunle [3] applied MLP to learn and exploit lagged correlations among S&P 500 stocks. Ahmet M. Ozbayoglu, Mehmet U. Gudelek and Omer B. Sezer [4] provide a state-of-the-art snapshot of the developed deep learning models for financial applications as of today. However, maybe due to the heavy computational burden of large deep learning models, lesser work has been done in terms of incorporating such models with massive recent trading behaviors. Here we followed the mean-reversion assumption and trading session design in [5] and the feature extraction protocol with established LSTM model in [6] and [7] gives the insight into handcrafted feature selection. We show that such prediction is indeed moderately possible, however when it comes to real trading environment, the speed of the training and the cost of the execution are of vital importance.

II. PROBLEM STATEMENT

The problem under consideration is to predict bid and ask price movement in near future based on high-frequency LOB data. More specifically, we use limit order book as input for feature engineering. As shown in Fig.1, it contains the flow of information at every time when National Best Bid and Offer changes.

DATE	TIME_M	EX	BID	BIDSIZ	ASK	ASKSIZ	QU_COND	QU_SEQNUM	NATBBO_IND	QU_CANCEL	QU_SOURCE	SYM_ROOT
20190102	10:00:00.077932022	Y	293.03	1	309.00	1	R	6215440	0	NaN	N	TSLA
20190102	10:00:00.078043277	Y	303.75	1	309.00	1	R	6215443	0	NaN	N	TSLA
20190102	10:00:00.078144663	Z	303.79	1	304.17	1	R	6215445	2	NaN	N	TSLA
20190102	10:00:00.078269542	K	303.57	1	304.54	1	R	6215446	0	NaN	N	TSLA
20190102	10:00:00.078381509	K	303.58	1	304.54	1	R	6215448	0	NaN	N	TSLA

Fig. 1: Order book list example

The information displayed includes the timestamp of the order book, the price, the volume, the sequence number, the symbol and so on. The interval between two events is one tick, the minimum incremental amount at which a trader can trade a security. The LOB works under specific matching rules of the stock exchanges. It accepts both limit orders and market orders. In the former case, the trader is willing to buy or sell the equities at a certain price. In the latter case, trading activity happens immediately by buyers actively reaching out to sellers, or vice versa. In fact, these unexecuted limit orders at different prices divide the LOB into levels. Stock exchanges denote National Best Bid and Offer (NBBO) as Level I quotes, meanwhile the Level II provides the depth of order book which generally refers to the

[1]Chenxi, Weiming and Yiyang are MQF students in Rutgers Bussiness School, Newark, NJ. Email: , wl489@scarletmail.rutgers.edu,

top ten bid orders and ask orders placed with their proposed prices and volumes. Due to the high price of Level II data, we only consider NBBO in our experiment. In addition, there is a renowned financial theory called mean reversion which suggests that the price of a stock tends to return towards its long running average price over time. When the price of a security has significantly moved away in some direction from its intrinsic value, we expect the market will pull it down in the opposite direction. This property can be seen from the oscillating behavior of the price around some intrinsic mean across the day in Fig.2. Here we made an assumption that a deep learning model can understand this swing nature of price movement to some extend.

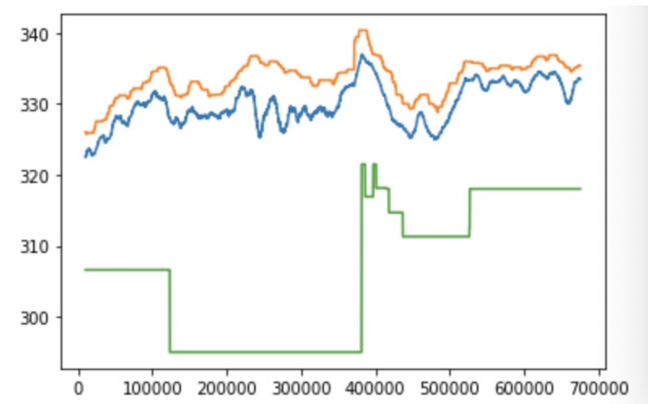

Fig. 2: A full-day bid and ask price snapshot of TSLA

We treat the prediction as a multivariate time series regression problem. With features extracted from limit order book, we conduct long-short term memory neural network to predict next tick bid and ask prices.

III. METHODOLOGY

The information contained in the snapshot of NBBO varies from tick to tick. Considering this, we utilize every events in the LOB in an online manner, namely, we use rolling window to extract features. Precisely speaking, the features are calculated every one thousand ticks with an overlap that just one tick will be excluded for next feature representation as shown in Fig.3.

Before the deep learning training process, the next step of the pre-processing is data normalization. We filtered all discernible outliers and conduct MinMax for the handcrafted features, as follows:

$$MM = \frac{X_i - X_{min}}{X_{max} - X_{min}}, i \in \mathcal{R}^N$$

Where N is the total sample size.

Many researches show that historical shocks remain its influence on the stock price in the future. The events in the LOB are likewise in a sequential manner. As we know, RNNs are the standard sequential system which means they are ideal for time series analysis. In our case, it's a many-to-one sequence problem as in Fig.4. We use gated RNN named LSTM as presented by Sepp

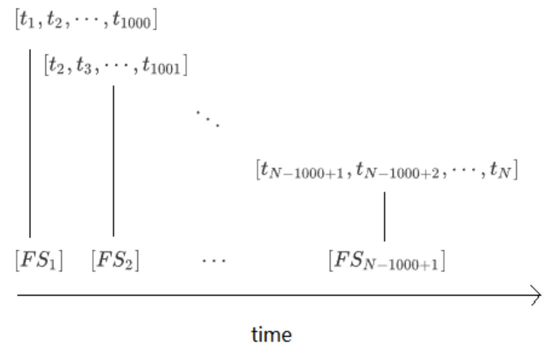

Fig. 3: zero lag feature extraction protocol

Hochreiter and Jurgen Schmidhuber [7] in 1997. The LSTM has been found extremly successful in many applications, such as unconstrained handwriting recoginition, speech recognition, handwriting generation and machine translation. The motivation to conduct LSTM in time series problems is it has been shown to learn long-term dependecies more easily than the simple multilayer perceptrons. For high-frequency trading, a full-day data is relatively long-term. Example of LSTM neural network can be seen in Fig.5.

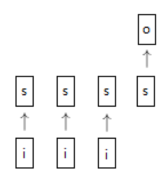

Fig. 4: A many-to-one sequence problem

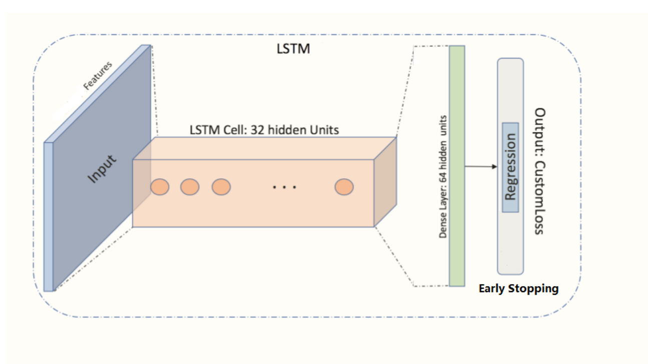

Fig. 5: LSTM model

IV. EXPERIMENTS & RESULTS

In this section, we provide results of the experiemnts we conducted. We performed the training process on Google Cloud Deep Learning VM with Keras. The server on which we ran the programs was Intel Xeon 24 cores, 2.2GHz processor with 156GB RAM and one Nvidia Tesla T4 GPU. The feature extraction by rolling window was running in multiprocessing.

Data pipeline: In this project, we obtained TAQ database of the NYSE from Wharton Research Data Services, all quote price for Tesla (ticker: TSLA) during the period 01/07/2019 – 01/11/2019, a total of 5 trading days. For the purpose of avoiding noises, we only consider the trading time 30 minutes after the market open and 30 minutes before the market close (between 10:00 am and 3:30 pm EST), only positive prices and volumes were considered (excluded whole row contains non-positive columns).

Feature pool: The original dataset contains best Bid price and best Ask price at a certain tick for Tesla, which can help us understand the trading behavior of the market. We selected 6 features from the original dataset, they are Date, Time, Bid Price, Ask Price, Bid Size, Ask Size and Bid Size and we calculated another 7 new features for building our model. Finally, we have 11 features in the dataset we used, and we normalize all the features into numbers between 0 and 1. The proportion among training, validation and testing sets is 6:2:2.

TABLE I: Feature Pool.

Features	Description
bid	Best quoted bid price
bid size	Bid size for the best quoted bid price
ask	Best quoted ask price
ask size	Ask size for the best quoted ask price
spread	The difference between best Bid and Ask
bid ask imbalance	Volume-weighted size difference
twap ask	Time-weighted ask price
twap bid	Time-weighted bid price
vwap ask	Volume-weighted ask price
vwap bid	Volume-weighted bid price

NN Structures: For LSTM, we have three hidden layers established in [6]. First is the LSTM layer with 32 hidden units, then a Dense layer with 64 hidden units and relu activation function and the last output Dense layer with 2 units. All weights were Xavier normal initialized. Biases were zero initialized. Adam optimization was used with the hyperparameters recommended in [8] and a base learning rate of 0.002, with mean squared error as the loss function. Additionally, we use 200 epochs and 32 mini-batch to train our model. Since heavily trained neural networks tend to be over-fitting, we use early stopping as our regularization method. By monitoring on validation loss we control the training time and achieve more robust results.

Results: Early stopping method stopped the training process at an early stage, the model reached its best performance with 0.7435 validation loss at the 7_{th} epoch in Fig.6. The accuracy on test set reaches 73.45%.

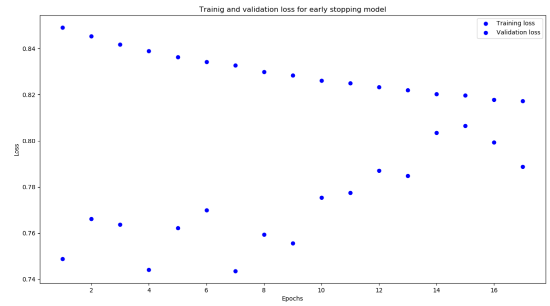

Fig. 6: Loss on training set and validation set

Fig. 7: Accuracy on test set

V. BACKTESTING

Our solution was formulated as a regression problem, so we initially utilized the traditional squared-error accuracy measure to evaluate our model. However, since our final predictions base on the majority of aggregated predicted changes for each tick, raw regression accuracy was not the best way to evaluate the model. We determined that a superior evaluation metric was to compare the average price change for ticks when the model makes correct predictions vs. the average price change for ticks when the model makes incorrect predictions (i.e., does the model correctly predict the largest price increases and decreases). Taking this approach, we developed a modified confusion matrix that considers the price changes predicted correctly, and price changes predicted incorrectly. We also looked at the average magnitude we predicted in either direction to see if we were predicting the majority of the largest swings.

Due to the trading costs and commissions existence in the real market, and we cannot apply the predicted price changes to our account while we buy cheap and sell high. We have to take a threshold to eliminate some unprofitable and uncertain trading chances, which means we only start our trading when the predicted price gap between the bid and ask is large enough to make a profit. According to the basic statistics, we believe that the bid-ask gap greater than 80% percentile, or less than 20% percentile of the historical data is profitable enough for our trading. When the gap is larger than the threshold, we order a higher price at the bid and place a lower price at the ask to do market making and make a profit by waiting for the gap to be extended, vice versa.

The strategy is quite simple but with a good performance with the help of the excellent predictive power of our NN-model. We successfully fetched trading opportunities of the market when the market is calling for liquidity or divergence. The strategy got as high as an 85% win rate in market making with a fantastic Sharpe Ratio. The novel performance shows that the NN model is quite suitable for high-frequency market-making as the data is sufficient, and nonlinear relationships between bids and asks are more comfortable to be revealed on this insufficient market.

VI. Conclusion

High-frequency trading strategy is not easy to access for every investors even as of today. Not only because tick data is difficult to process in real time, but also it cannot simply be explained by singular linear model or non-linear model. Some investor with fortune to get access in this field in the earlier days could use simple mean-reversion or linear regression to make money in this market. Whereas this ‘good old time’ had passed and the opportunities to trade become smaller because recently more and more investor reached to this market and plan to use multiple ways to conquer high frequency market. Every high-frequency trader is seeking for an efficient model to explain tick data. Tick data is hard to explain not only because it’s raw and uncompressed data of the trading behavior of the market (only contains Bid/Ask Price and Size), but also, it’s time series data with strong self-correlation. However, in this paper we discovered LSTM has ability to capture the spread changes in time series. First of all, calculating new features for modeling would help digging deeper into discovering trading signal behind the tick data; selecting new features that are backed by financial and statistical theory become essentially important. Selecting features appropriately and choosing model correctly are two significant keys that get high accuracy rate, even naïve LSRM has epically good performance in the prediction, we reached 73% prediction accuracy on our test set. Based on the prediction for next tick, a strategy, long/short when prediction is higher/lower than real price, outperformance the market. This strategy has 2 thresholds to reduce trading frequency, which is a way to control the risk. Clearly this strategy is still in a very beginning stage, and an important aspect of the proposed pipeline is availability of enough computational power to keep updating the model weights in real time. Nonetheless it points out a good direction for us to put more research and study. In conclusion, using LSTM model on high-frequency trading strategy is a great inspiration on building strategy and it’s really worth to put more effect on it.

VII. Future Work

There are lots of work we can do to improve our model in the future. For feature engineering, we can eliminate some unnecessary properties and fetch higher ranked features by using Stacked Denoising Autoencoder(SDAE). With the help of SDAE, we can clear some noise and help the network using more useful features. To get the higher ranked features, we can use the weights of the layer before the output layer in SDAE, which contains higher-ranked information of features. What’s more, we can take some other instruments’ data as features of our target as some industrial or economic factors usually interconnect them.

For instruments that have no sufficient data, we still can apply our framework on it by setting the initial weights of the network to pre-trained network weights, which called transfer learning. As we all know, investors’ behaviors are quite similar, no matter what instruments they are trading. So it is meaningful and helpful to do some transfer learning from some long-lasting instruments like AAPL or MSFT. We can even go further by using the BERT framework on our task. The Bidirectional NN can discover the relationship between targets and features bidirectionally. We can pre-train our NN in BERT way and transfer the optimal weights as initial weights to our productive model. We can even mask some historical data points to create some training environment to do better pre-training.

There are also some arguments on applying attention to finance, especially on quantitative trading. By a famous paper *Attention is All you Need*, people found that attention may effectively replace the role of recurrent generators. The shape of the trading volume in one day is like a U, which means investors pay more attention to the beginning and end of the intraday market. This phenomenon is perfect for an attention model.

References

[1] J. W. L. I. P.-A. M. Hassan Ismail Fawaz, Germain Forestier, “Deep learning for time series classification: a review,” *Data Mining and Knowledge Discovery https://doi.org/10.1007/s10618-019-00619-1*, vol. 33, pp. 917—-963, 2019.

[2] K. N. Masaya Abe, “Cross-sectional stock price prediction using deep learning for actual investment management,” *URL https://arxiv.org/abs/2002.06975*, 2020.

[3] G. I. Ben Moews, “Predictive intraday correlations in stable and volatile market environments: Evidence from deep learning,” *Physica A: Statistical Mechanics and its Applications https://doi.org/10.1016/j.physa.2020.124392*, vol. 547, 2020.

[4] O. B. S. Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, “Deep learning for financial applications : A survey,” *arXiv preprint arXiv:2002.05786*, 2020.

[5] P. R. Prakhar Ganesh, “Deep neural networks in high frequency trading,” *arXiv preprint arXiv:1809.01506*, 2018.

[6] J. K. M. G. A. I. Adamantios Ntakarisa, Giorgio Mironeb, “Feature engineering for mid-price prediction with deep learning,” *IEEE Access https:/doi.org/10.1109/ACCESS.2019.2924353*, vol. 7, 2019.

[7] C. J. L. Martins, “Market and limit orders and their role in the price discovery process,” *THE JOURNAL OF FINANCE http://faculty.haas.berkeley.edu/hender/IIROC.pdf*, 2019.

[8] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.