Benchmark Dataset for Short-Term Market Prediction of Limit Order Book in China Markets

Charles Huang, Weifeng Ge, Hongsong Chou, Xin Du

FinAI Laboratory
Hong Kong Graduate School of Advanced Studies

Charles Huang is the corresponding author, and a Professor; Weifeng Ge is an Assistant Professor; Hongsong Chou is a Professor; Xin Du is a lecturer; all with the FinAI Lab of Hong Kong Graduate School of Advanced Studies. The authors can be contacted at emails: <firstname>.<lastname>@gsas.edu.hk

November 30, 2020

ABSTRACT

Limit Order Book (LOB) has generated "big financial data" for analysis and prediction from both academic community and industry practitioners. This paper presents a benchmark LOB dataset of China stock market, covering a few thousand stocks for the period of June to September 2020. Experiment protocols are designed for model performance evaluation: at the end of every second, to forecast the upcoming volume-weighted average price (VWAP) change and volume over 12 horizons ranging from 1 second to 300 seconds. Results based on linear regression model and state-of-the-art deep learning models are compared. Practical short-term trading strategy framework based on the alpha signal generated is presented. The data and code are available on Github (https://github.com/HKGSAS).

***K*eywords** High-Frequency Trading · Limit Order Book · Artificial Intelligence · Machine Learning, Deep Neural Network · Short-Term Price Prediction · Alpha Signal · Trading Strategies · China Stock Market

Key Findings

- There is a gap of open benchmark high frequency LOB dataset and model for researchers to objectively assess prediction performances, which this paper serves to bridge.
- The typical feature set utilized by many researchers is lacking. The authors propose a more effective feature set capturing both LOB snapshot and periodic data.
- The prediction target is similarly too simplistic in published literature -- mid-price change direction for the next few events, not suitable for practical trading strategy. The authors propose to predict the price change and volume magnitude over 12 short-term horizons.
- This paper proposes to compare the performance of baseline linear regression and state-of-the-art deep learning models, based on both accuracy statistics and trading profits.

1. Introduction

High frequency trading is a competition of speed and strategy, and has grown into a field dominated by a few large players such as Citadel Securities, Virtu and Two Sigma. In Q1 2020, the publicly listed HFT firm Virtu, with estimated US equity retail trading market share of 27%[1], reported a jump in trading revenue to $1 billion, aided by the coronavirus induced market volatility. This implies a market size of around US$4 billion in Q1 2020 for US retail equity HFT alone.

Market makers utilize their edge to provide liquidity, and make a profit from the bid-ask spreads they quote in the market, which generally benefit from higher volatility and wider spreads. Institutional investors use algorithmic trading, a form of high frequency trading, to execute their trades, mostly to minimize the market impact. Hedge funds explore high frequency data to seek alpha in their trading strategies.

Limited Order Book (LOB) is the main trading match engine adopted by almost all the major exchanges worldwide. LOB is the true heartbeat of the financial markets -- most participants are not concerned or aware of its importance or even existence, but LOB is responsible for pumping multi-trillion USD of trades ever day in the financial markets around the world.

LOB is quite an interesting dynamic system resulting from competitive behavior of a large and diverse range of market participants. LOB is indeed a very challenging problem from the research perspective. Many academic researchers [1,3] have tried to model the behavior of LOB over the past few decades. Economists tend to model LOB from the agent behavior perspective, and to model the decision making process of different individual/type of market participants, assuming a goal of maximizing of their utility. Physicists have done substantial modeling of LOB as well, typically from a stochastic and statistic modeling angle. Zero knowledge model has been used as a simplified case to model LOB, with more advanced models utilizing Hawkes processes. Researchers have had some success in explaining the stylized facts observed from LOB: long tails in returns (not a Gaussian distribution as in most financial data), clustered volatility, but have faced significant challenges in modeling LOB dynamics, probably due to the high dimension of the LOB data resulting from an evolving and competitive market behavior. With the recent rapid advancement in deep neural networks, more computer scientists [2,7,8] have proposed to analyze LOB with Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) such as Long-Short Term Memory (LSTM).

China HFT market has presented interesting opportunities and challenges. During the high volatility times in 2015, the stock market dropped 40% within a few months. Citadel had utilized their advanced HFT technology and strategy, and profited handsomely from HFT. However, the rapid fire trading has caught the eyes of the Chinese regulators, resulting in account suspension, a lengthy investigation and a fine of US$97 million. The exchanges have since tightened measures against alleged market manipulation.

There are very few public benchmark LOB datasets for researchers to analyze and develop their models [6]. Some public datasets are either outdated or limited in coverage (only forex, a few stocks or a short period of time). This paper introduces a publicly available benchmark dataset[2]

---

[1] Virtu's SEC Rule 605 "retail" marketable US equities share volume executed was approximately 450 million shares per day in 2020Q1, as compared with total Rule 605 "retail" volume of 1.7 billion shares per day, and 9 billion total consolidated volume per day.

[2] The dataset is supplied under China stock exchanges' promotion of Level-2 data adaptation scheme, and is for academic research purpose only.

for China stock market, consisting of 2270 stocks, 8 billion order and trade messages, covering the period of June to September 2020. We have cleaned, preprocessed the data, constructed LOBs, and annotated fields for easy supervised learning. Because of the large volume of original tick-level data, we have chosen to deliver the post-processed LOB benchmark data in two forms: a smaller set covering 20 stocks of size 2.6 GB which is open for anybody to download and study, and a complete set available for serious researchers upon request. We have also made the Python code used for LOB model projection available on github (https://github.com/HKGSAS) -- researchers and practitioners are welcomed to collaborate with the FinAI Laboratory at Hong Kong Graduate School of Advanced Studies (GSAS).

We propose to reconstruct LOB into a richer feature set X -- for each stock at the end of every second, LOB is rebuilt into two components: the *snapshot* component and the *periodic* component. The snapshot component incorporates 10-levels of market data on both the bid and ask sides, including price, size, number of orders, weighted average of order staleness. The periodic component covers the market activities between two snapshots, e.g. from the LOB of a second ago to the current LOB. The periodic component includes three parts, the *standard part*, consisted of VWAP price, trade volume, number of trades, previous close, open, high, low, close price; the *executed trade direction/size part*, consisted of (buy | sell) (total | large | medium) price, volume, number of orders, and average staleness on the passive side; and the *cancellation part*, consisted of (buy | sell) LOB level at cancellation time, LOB level at order placement time, number of canceled orders, average price of canceled orders, size of canceled orders, and staleness of canceled orders.

For the Y-Label (prediction target) part, we propose to forecast not just the mid-price change direction in the next events as in almost all literature, but to forecast the VWAP price changes and volume for the next 1, 2, 3, 5, 10, 20, 30, 60, 120, 180, 240, 300 seconds. While the traditional mid-price direction prediction experiments are of theoretical value, they are far from practical HFT strategy development. To facilitate efficient supervised learning, we have annotated labels for prediction tasks in the benchmark dataset.

Based on the alpha signals generated by successful models, HF traders can develop trading strategies based on the horizon of his/her choice. Ultra-fast (with horizon within 5 seconds) traders should investigate on how to shorten delay of the whole trading cycle -- receiving market data from exchanges, processing the data to make trading decisions (need fast models), managing position/risk, and sending/monitoring/updating orders sent to the exchanges. Short-term traders, with holding horizon within 300 seconds, can benefit greatly if their models can generate correct alpha signals on a consistent basis. Long-term investors could benefit from better trading execution by incorporating short-term alpha models.

The rest of the paper is organized as follows. We present the benchmark dataset and experimental protocol for the China stock market in Section 2. In Section 3 we describe the engineering of our benchmark models, including a linear model, nonlinear MLP (MultiLayer Perceptron) model, and CNN / LSTM based state-of-the-art models. In Section 4, we compare the model results, including short-term alpha trading strategy results based on model predictions. Section 5 gives the conclusion and future research directions.

2. The Benchmark LOB Dataset

In this Section, we present the benchmark dataset for the China stock market, detail the process of raw data cleaning and LOB reconstruction, propose a more refined feature set and more realistic prediction labels, and design both statistical and trading driven experimental protocols to analyze this benchmark LOB dataset.

2.1. China Stock Market Briefing

There are two stock exchanges in China, one in Shanghai and one in Shenzhen. Historically, stocks listed on Shanghai tend to be larger and to be controlled by the state, while stocks listed on Shenzhen tend to be smaller and to be privately controlled. The distinction however has been diminishing recently with Shanghai launching a new Sci-Tech innovation board (STAR Market) in 2019. While trading rules are similar on the two exchanges, Shenzhen provides more detailed tick-level market data, i.e. each order submission and each execution, while Shanghai provides not-as-fine granular data every 3 seconds. The major distinct trading rules from other markets are: daily trading price limits of plus and minus 10% from the previous day's closing price (with some exceptions for newly listed stocks, recently introduced STAR Market, high risk "ST" stocks, etc.) and T+1 trade/settlement, entailing one can NOT buy and sell the same stocks in the same day. There is a sell-side stamp duty of 0.10% (long term holders are exempt) and brokerage fee is approximately 0.03% for both buy and sell side. So the round trip transaction costs amount to 0.16% or 16 bps. The minimum price tick is 1 cent for stocks, and the minimum buy order is 1 hand (100 shares).

There are around 4000 stocks listed on the two exchanges, with a market cap of US$9 trillion, equivalent to approximately 2/3 of Chinese GDP. The stock listing process has been mostly regulatory approval driven and there have been strict profitability requirements. As a result, many Chinese internet and high growth companies (Alibaba and Tencent) have chosen to list offshore, giving the domestic exchanges a nudge to adjust the listing process toward the more common disclosure-based international norm. Because the vast majority (80%) of stock trading have been retail, the regulators feel stronger responsibility to ensure the well being of the markets. To the investors' dismay, the Chinese stock market performance (3x increase) since its establishment in the early 90s has been only a small fraction of the astonishing growth in the Chinese GDP (35x increase). Most of the funds driving the economic growth have been supplied by the state controlled commercial banks.

2.2. Tick-Level Data Description

Only Shenzhen provides tick-level messages, detailing each order, cancel and trade. Hereby, we choose all stocks listed on Shenzhen Stock Exchange as the basis of our benchmark dataset. We have received the raw Level-2 data supplied by the exchange covering the period of June 2020 to September 2020 in two data streams: the *order stream* and the *trade stream*. The order stream is a flow of messages in binary format, each consisted of the following fields: message standard header, channel number[3], order sequence number(id), market data type, security id, security exchange, order price, order quantity, order buy/sell direction, time (in millisecond), and order type[4]. The trade stream message consists of the following fields: message standard header, channel number, trade sequence number(id), market data type, bid-side order id, offer-side order id, security id, security market, trade price, trade quantity, trade type[5], and time (in millisecond).

Note that the time stamp is from the exchange, not the message receiving time by a trader, which can be delayed by around 50 ms (co-location) to 300ms (the same city with fast broadband access). There is no hidden order allowed. For every trading day, open auction period is 9:15-9:25am (China Standard Time or GMT + 8 hours); continuous trading periods are 9:30-11:30 and 13:00-14:57 with a noon break 11:30-13:00; close auction period is 14:57-15:00.

The raw data we used is approximately 800 GB, making it one of the largest benchmark financial datasets. We have cleaned the raw data to make sure that irrelevant (administrative) messages are discarded, all the fields are within reasonable range and follow prescribed logic (e.g. the

[3] Channel number is assigned by the exchange based on the matching engine handling a particular stock.

[4] 1-Market order; 2-Limit order; 3-Peg to the best price of my side

[5] 0-Completed; 1-Cancelled

sequence numbers are strictly increasing with time). The raw data is further processed into LOBs with feature sets and labels.

2.3. LOB Reconstruction

Note that almost all published research papers have used the standard output files supplied by LOBSTER (https://lobsterdata.com/info/DataStructure.php) [4]. LOBSTER outputs two files: message and orderbook, with an user specifying input parameters such as the number of events between two consecutive LOB snapshots and the number of LOB levels in a snapshot. Because the message file is raw and large, researchers routinely dropped this information-rich file and gravitated toward the neat orderbook file, containing a simplistic snapshot of typical 10-level deep price and volume data on both ask and bid sides.

Some researchers[5] have added more statistics of the simple orderbook data in the LOB reconstruction process. This 144-dimension LOB representation was created when Support Vector Machine was the state-of-the-art machine learning technology. The "time-insensitive" parts -- spread and mid-price, price differences, and accumulated differences -- are simple arithmetic operations on the existing price and volume data, which have been rendered redundant in the Deep Learning era. The "time-sensitive" parts -- price & volume derivation, average intensity per type, relative intensity comparison, and acceleration -- are explicit speed and acceleration measurements of price and volume, which can be readily modeled by Deep Learning models such as RNN/LSTM.

Our aim is to solve the practical problem -- the short-term price change and volume prediction, hence we are extracting as much useful LOB information as possible from the whole message file, not overlooking important data for the sake of modeling simplicity. In addition to price and volume data at a LOB level within a snapshot, we distill more useful data such as how concentrated and how stale the orders are, because order concentration aligns with information concentration and order staleness indicates information staleness. Moreover, the actual trade and cancellation data, the final scorecard of the market, should give as much if not more information on the current status of the market and possible future directions. Puzzlingly, LOB researchers have not analyzed or utilized the trade and cancellation data as much as the snapshot data.

To fulfill the pursuit of extracting all useful information to represent LOB dynamics, we have chosen the more robust path of writing our own data processing module (a C++ program), recreating the trading time-line by matching all order and trade messages for every stock, and "compressing" the raw data into two components: the *snapshot* component and the *periodic* component.

The snapshot component includes not only the 10-level depth of price and size data on both the bid and ask side, but also the *number of orders* and the *weighted average staleness* of the orders at every level. The number of orders details the order concentration. The weighted average staleness (the difference between order submission time and LOB snapshot time) describes the urgency and information freshness of the orders. Exhibit 1 demonstrates an example of the snapshot component of 000001.SZ stock at the end of [time]. Interested readers can run our open source code online (https://github.com/HKGSAS) to play a time-lapsed movie of LOB snapshot overtime.

[]

Exhibit 1: LOB snapshot component example

The periodic component covers the market activities between two consecutive snapshots (default set as one second), including three parts, the *standard* part, consisted of volume-weighted average price, trade volume (can be 0), number of trades, and prices including previous close, open, high, low, close (all prices will be set the same as the previous close price if there is no trade in the period); the *executed trade direction/size* part, consisted of (buy | sell)[6] (total | large | medium)[7] average price, volume, number of orders[8], and the average staleness of the passive side; and the *cancellation* part, consisted of canceled (buy | sell) average order book depth at cancellation, average order book depth at placement, average price, volume, number of canceled orders, and average staleness. The standard part is essentially the candle-stick bar widely used by traders and analysts in many markets. The executed trade direction/size part gives insight into the imbalance between the buy-side and sell-side initiated trades, e.g., large, concentrated buy-side initiated trades against small, distributed and stale sell-side orders might foretell a strong buying momentum and vice versa. The cancellation part offers insight in the possible motivation: giving-up or spoofing.

Exhibit 2 shows the LOB periodic component of stock 000001.SZ for the day of [] at 30 minutes increments. Interested readers can run our open source code online (https://github.com/HKGSAS) to interactively check the periodic component of any stock for any period and any increments.

[]
Exhibit 2: LOB periodic component example

We have reconstructed the LOB for the period (01-Jun-2020 to 30-Sep-2020) at the end of every second from 9:15:01-11:30:00 and 13:30:01-15:00:00 on every trading day for every selected stock. To follow the philosophy of providing all the data critical for a good prediction, we have chosen to include the open (9:15:01-9:25:00) and close (14:57:01-15:00:00) auction periods, since these periods provide a wealth of information to traders. We use trading phase[9] to indicate the status of a stock during the second.

To our knowledge, the benchmark LOB data presented in this paper is the first to utilize more refined snapshot data and detailed periodic data as important input for forecast model. In a separate paper, we will compare the information content contained in different LOB features, and their effectiveness in predicting short-term price change and volume.

2.4. **Feature Set - X**

Constructing the right feature set (X) is crucial in building an accurate and robust forecast model. While many earlier academic research were focusing on proving or disproving the market efficiency hypothesis by testing the predictive power of some *simple* factors (more commonly referred as features in neural network research), such as historical returns, order book imbalance,

---

[6] We use the more recent side of order to represent the direction of a trade, e.g. if the buy-side order is more recent than the matched sell-side order, the trade is marked as a buy-trade.

[7] We choose certain threshold traded share numbers to categorize trade into: *large* (> 10,000), *medium* (> 3,000), otherwise *small*. Other adaptive methods based on actual historical trading share numbers can be used as well.

[8] When a trade is matched as one order on one side with many orders on the opposite side, the order number count is 1 while the trade number count is many.

[9] Trading phase can be of the following status: 0-Pre-open; 1-Open auction; 2-Between the end of open auction and continuous trading period; 3-continuous trading period; 4-Noon break; 5-Close auction; 6-Market closed; 7-Afterhours trade; 8-Temporary trading halt; 9-Fuse break;

order flow imbalance, and exponential moving averages of these time series data. In the age of neural network driven modeling, researcher have utilized more complex feature set, in line with better computing power (especially with specialized chips such as GPU and TPU) and more sophisticated models.

A commonly used LOB feature set by many researchers [5] is a 144-dimensional representation, consisted of the basic part (ask and bid price & volume at 10-levels), time-insensitive part (some simple statistics such as spread, mid-price, price & volume means), and time-sensitive part (absolute and relative velocity of different items). Some have added various types of technical indicators attempting to capture momentum and mean-reversion trends observed in the markets.

In this paper, we propose to utilize a richer (but lower dimension) LOB feature set reconstructed directly from the order and trade messages, with the an emphasis on potential usefulness for forecast vs. model neatness. Our feature set is a 124-dimensional array, grouped into two parts: the 1st part contains the transactional data from the past period (default set at 1 second) and the 2nd part contains data contrasting the buy vs sell forces, organized in two symmetric channels for easy CNN feature extraction.

The first (transactional) part is of dimension 8, consisting of data extracted from the LOB periodic part: VWAP price, volume, number of orders, and five candle-stick bar type prices (previous close, open, high, low, close). The second part is of dimension 116, and can be organized into two channels, buy and sell. Each channel is of dimension 58, and contains data extracted from LOB snapshot and periodic parts: the ten-level depth LOB data of dimension 4x10 = 40: 4 being price, size, number of orders, and average staleness of orders; the executed trade part of dimension 3x4 = 12: 3 being total | large | medium, and 4 being price, volume, number of orders and staleness of the passive side; the cancellation part of dimension 6, being average market depth at cancellation time vs. at placement time, average price, volume, number of cancellations.

To facilitate efficient machine learning, we proceed to normalize the input data **X** as following:

- Prices are left as is. In the case of no trading volume in a period, the corresponding prices are set the same as the previous closing.
- Volumes are normalized by dividing by the top 10% volume quantile level, but limited to 1.
- Number of orders are normalized similar to volumes, dividing by the 10% order number quantile level, but limited to 1.
- Staleness is converted into three categories: 0 (5 seconds or less), 1 (between 5 and 30 seconds), 2 (longer than 30 seconds).
- Cancellation part is normalized with data from cancellations (not from trades as above).

2.5. **Label - Y**

Most published research set Y as the mid-price movement direction upon the next change or the next events. This simple label, while easy to understand, is of limited practical usage. We have set up more challenging and useful forecast targets, i.e. predicting the upcoming average price change and volume for various time horizons t seconds, ranging from 1 second to 300 seconds -- we have chosen 12 t in our study: 1, 2, 3, 5, 10, 20, 30, 60, 120, 180, 240, 300 seconds. For prices, we first calculate logarithmic changes and then discretize into 5 quantiles of different sizes -- 10%, 20%, 40%, 20%, 10%, with corresponding labels of -2, -1, 0, 1, 2. The more granularized tail measurement guides prediction model to be more sensitive to the extremes

versus the not-much-changed middle portion. Alpha generating models successful at predicting the "tails" will likely generate out sized returns.

$$\ln(P_t / P_0) \subseteq \{-2,-1,0,1,2\}$$

For volume, we discretize the normalized data into 3 quantiles of different sizes -- 20%, 40%, 40%, with corresponding labels of 0, 1, 2. The rationale is to distinguish low, normal and high volume of trading.

$$V_t \subseteq \{0,1,2\}$$

Since we have already reconstructed LOBs at the end of every second, the **Y** label can be calculated in a straight-forward way: for a horizon of t seconds, we can simply aggregate the price and volume levels for the next t LOBs. For example, the upcoming 1-second horizon Y (price change and volume) can be fetched directly from the LOB ending at the next second. The upcoming 300-second Y (price change and volume) information can be aggregated from the next 300 LOBs.

For the training set data between June and August 2020, we compare the quantile thresholds across the 12 horizons in Exhibit 3. Note that the top 10% quantile price change cut off level increases from [] bps for 1-second horizon to [] bps for 60-second horizon to [] bps for 300-second horizon.

{Bar chart, x - 12 horizons from 1 - 300 not in scale, y - bps change for price, each bar cut in five parts, in different cue, top "green" for increase, bottom red for decline. The goal is to show how much the range increases with time}

Exhibit 3: Quantile range comparison for 12 horizons.

2.6. Experiment Protocol

With feature set - X and label - Y setup, we proceed to define the experimental protocol for this paper. We use the continuous trading (9:30 to 11:30 and 13:00 to 14:57 each trading day) data from June 3 to August 31 for in-sample training and validation, and the data from September 1 to 30 for out-of-sample testing.

To train a model, we use supervised learning technique by feeding in the 124-dimension X-feature set of the most recent 10-seconds (10x124 dimension) as input and a Y-label to be predicted. For each of the 12 horizons ranging from 1 to 300 seconds, we train one model to predict the price change, and one model to predict volume. For one experiment (such as training a linear model), we will train 12*2 models. Cross-entropy loss is used as the loss function.

We measure model performance on the out-of-sample data from September 1 to September 31 from two perspectives: 1) traditional statistics such as absolute and normal prediction error on price change and volume; 2) profit, success-rate, accumulated return of a model-driven trading strategy.

To calculate the traditional model measurement for a horizon t seconds, at the end of each second of a trading day, a model is applied to predict the price and volume change for the

upcoming *t*-second horizon, and the model performance is measured with the standard accuracy, recall, and F-measure statistics.

More interestingly, we measure a model's performance based on its effectiveness on a simple day-trading strategy. For a given stock, and at the end of every second from 9:30:00 to 11:25:00 and 13:00:00 to 14:25:00, we use the model to project the the (price change, volume) pair for 5-second and 1-minute horizons, and execute trades based on model predictions:

- Up-moving market: if the projected price for 1-minute horizon is 2 categories higher than the price for 5-second horizon, and the projected volume for both horizons are not in the bottom 0 category, we buy the stock at the realized average price for the upcoming 5-second for the amount of RMB 10,000, and sell at the realized average price of 5-60 seconds, assuming a round-trip transaction costs of 20 bps.
- Down-moving market: if the projected price for 1-minute horizon is 2 categories lower than the price for 5-second horizon, and the projected volume for both horizons are not in the bottom 0 category, we sell the stock at the realized average price for the upcoming 5-second for the amount of RMB 10,000, and buy back at the realized average price of 5-60 seconds, assuming a round-trip transaction costs of 20 bps.
- Otherwise, don't trade.

Note that we made some assumptions: we can and will always open and close a trade within the minute; we are not constrained by capital and stock availability to short.

Then we measure the performance of this simple day-trading strategy: number of completed trades (open and close) executed a day and over the test period, the average return per trade, the distribution of the returns. The accumulated return is also compared.

3. The Benchmark Models

The main purpose of this paper is to present the high frequency trading benchmark dataset for researchers to develop their prediction models and trading strategies. The benchmark models are selected for their representativeness and easiness to replicate (all code can be obtained at https://github.com/HKGSAS). We encourage researchers to develop more sophisticated time series models, especially by utilizing the latest advancement in deep learning, to better predict price change and volume, and to explore profitable trading opportunities.

A linear multinomial logistic regression (LR) model is shown first. Five neural network based models, including a few state-of-the-art models, are presented: a MultiLayer Perceptron (MLP) model, a shallow Long-Short Term Memory (LSTM) model, a Convolutional Neural Network (CNN) model, and a hybrid CNN-LSTM model. The models and results presented in this section are *universal* models based on data across all stocks.

3.1. Logistic Regression

Linear models are known for its simplicity and interpretability. L2 regularization methods have been used to reduce model complexity. We have implemented the Multinomial Logistic Regression with PyTorch. The detailed configuration of this linear model is shown in Exhibit 4: consists a single layer perceptron taking the past 50 seconds of 124-diminsional X as input (flattened to 50*124 = 6200) and a softmax layer to output one of target categories. As in binary Logistic Regression, the one with the maximum likelihood is assigned with the category label.

The softmax activation is applied as follows:

$$\text{softmax}(x_i) = \frac{e^{x_i}}{\Sigma_j e^{x_j}}$$

Logistic Regression (31000 parameters)
Input @ [1x6200]
Dense @ 5 Units (Softmax activation)

Exhibit 4: Multinomial Logistic Regression model architecture

3.2. MultiLayer Perceptron

MultiLayer Perceptron is a straight forward but versatile deep neural network which can approximate complex non-linear relationship between inputs and outputs. MLP does not assume any preconceived temporal or spatial relationship in the input data, but can "automatically" explore to find such connections. The drawback of MLP model is the number of parameters involved at 4.8 million, and the difficulty in explaining the causal effects. MLP serves as a strong baseline deep learning model for the LOB prediction task. The configuration of the input, loss function, training steps and optimizer is set the same with that in Logistic Regression.

MultiLayer Perceptron (4813120 parameters)
Input @ [1x6200]
Dense @ 512 Units
Dense @ 1024 Units
Dense @ 1024 Units
Dense @ 64 Unit
Dense @ 5 Units (Softmax)

3.3. Shallow LSTM

Researchers have observed the temporal relationship between LOB features and subsequent price movements. RNN models, in particular LSTM models, have been extensively deployed to model many sequential relationship between inputs and outputs, including machine translation and time series analysis. LSTM models incorporate the attention mechanism with different gates (input gates, forget gates, output gates) to ensure the long term dependencies between different states can be captured while reducing the gradient vanishing problem. LSTM models form part

of state-of-the-art neural network architecture for any time series modeling.

In the shallow LSTM model shown in Exhibit 5, the input is 50 consecutive history LOB states represented as 50x124 matrix and the output are the prediction results in the subsequent states.

LSTM (129704 parameters)
Input @ [50x124]
LSTM @ 124 Unit
Dense @ 5 Units (Softmax)

Exhibit 5: Shallow LSTM architecture

3.4. **Convolutional Neural Network**

Convolutional neural network (CNN) has been widely used in computer vision and machine learning for its ability in capturing spatial dependencies among adjacent pixels. Individual cortical neurons respond to stimuli only in a local restricted region called receptive field. By concatenating t (=50) adjacent LOB sates, we get a 2D LOB price moving images. CNN is then designed to process the LOB data, modeling the contextual relationship among different states and different dimensions. The CNN model architecture is shown in Exhibit 6.

Convolutional Neural Network (568704 parameters)
Input @ [50x124]
Conv (+ BN + ReLU) 4x124 @ 64; 1x1 @ 128; 4x1 @ 256;
MaxPool 2x1 @ 256 (stride = 2x1)
Conv (+ BN + ReLU) 3x1 @ 512
MaxPool 2x1 @ 256 (stride = 2x1)
GlobalAvgPool 8x1 @ 512
Dense @ 5 Units (Softmax)

Exhibit 6: CNN architecture

3.5. **CNN-LSTM**

LSTM models are powerful in processing the sequential signal for capturing long term and short term relationship between different states. CNN models possess strong ability in capturing spatial characteristics in local receptive fields. By combining CNN with LSTM models, we can utilize CNN to extract discriminative features and feed to LSTM to explore relationships between history states. The CNN-LSTM model architecture as shown in Exhibit 7 is quite similar to the CNN model, except that the GobalAvgPool layer is replaced by a LSTM Layer with 124 units.

CNN-LSTM (717056 parameters)
Input @ [50x124]
Conv (+ BN + ReLU) 4x124 @ 64; 1x1 @ 128; 4x1 @ 256;
MaxPool 2x1 @ 256 (stride = 2x1)
Conv (+ BN + ReLU) 3x1 @ 512
MaxPool 2x1 @ 256 (stride = 2x1)
LSTM @ 64
Dense @ 5 Units (Softmax)

Exhibit 7: CNN-LSTM architecture

4. Model Results

Because the relatively large data set and the extensive resources required to conduct full experiments, we have proceeded to analyze a smaller dataset of 20 actively traded stocks over three prediction horizons: 5, 60 and 300 seconds as the first step, which is easier for researchers to replicate. The machine learning models are implemented with PyTorch and Nvidia RTX 2080 Ti GPUs. The following training setup have been used for reference:

- *Training and Test Configuration*: The first three month data (June-August 2020) are used to train and the last month data (September 2020) are used to test. There are 17362482 training samples and 6109880 test samples in this experiment.
- *Max Iteration and Learning Rate*: The training set are run for 10 times (10 epochs). The initial learning rate is set to 0.01, and then multiplied with a scaling factor 0.1 for every 4 epochs.
- *Objective Function and Labels*: The LOB price labels are quantized into five categories of different size (from top to bottom [10%, 20%, 40%, 20%, 10%]) as described in the previous sections. The cross entropy loss is used for model parameter learning.
- *Optimization Solver*: SGD optimizer is used to update the model parameters. The momenta is set to 0.9, the weight decay is set to 0.0005. The L2 penalty is used in every model for parameter regularization.

To test the prediction performances of different models, we adopt commonly used evaluation metrics for multi-label classification, including Averaged Accuracy, Weighted Accuracy, Weighted Recall and Weighted F-score. The weighted metrics take the class imbalance into consideration and output more balanced indicators. The Precision, Recall, F-Measure for each quantile is presented for comparison. The multi-class correlation – Cohen's Kappa are also computed. Exhibit 8 details the performance statistics for the selected models: LR, MLP, CNN, LSTM and CNN-LSTM.

	Logistic Regression			Multilayer Perceptron			CNN			LSTM			CNN-LSTM		
	H5	H60	H300	H5	H60	H300	H5	H60	H300	H5	H60	H300	H5	H60	H300
Averaged Accuracy	0.23	0.25	0.24	0.28	0.29	0.24	0.29	0.25	0.25	0.29	0.26	0.25	0.30	0.27	0.25
Weighted Accuracy	0.44	0.38	0.51	0.66	0.50	0.36	0.73	0.55	0.49	0.73	0.75	0.57	0.62	0.49	0.37
Weighted Recall	0.33	0.33	0.35	0.41	0.39	0.32	0.43	0.37	0.35	0.43	0.41	0.38	0.42	0.38	0.32
Weighted F-Measure	0.37	0.35	0.40	0.47	0.42	0.34	0.51	0.42	0.40	0.51	0.50	0.44	0.47	0.41	0.34
Precision quantile [0,0.1]	0.06	0.10	0.10	0.08	0.24	0.14	0.05	0.14	0.16	0.05	0.11	0.14	0.10	0.16	0.16
Precision quantile [0.1,0.3]	0.16	0.26	0.19	0.17	0.19	0.24	0.16	0.13	0.28	0.16	0.13	0.15	0.20	0.20	0.19
Precision quantile [0.3,0.7]	0.61	0.52	0.70	0.83	0.69	0.54	0.89	0.74	0.66	0.89	0.88	0.75	0.80	0.68	0.56
Precision quantile [0.7,0.9]	0.27	0.27	0.09	0.25	0.17	0.17	0.29	0.13	0.04	0.29	0.05	0.11	0.30	0.18	0.15
Precision quantile [0.9,1.0]	0.04	0.08	0.11	0.07	0.16	0.12	0.05	0.09	0.09	0.05	0.11	0.12	0.08	0.14	0.17
Recall quantile [0,0.1]	0.13	0.23	0.21	0.26	0.27	0.15	0.31	0.22	0.21	0.31	0.34	0.23	0.26	0.27	0.18
Recall quantile [0.1,0.3]	0.24	0.25	0.24	0.33	0.32	0.24	0.36	0.28	0.24	0.36	0.33	0.30	0.34	0.30	0.24
Recall quantile [0.3,0.7]	0.39	0.42	0.42	0.42	0.44	0.43	0.42	0.42	0.43	0.42	0.42	0.43	0.43	0.43	0.43
Recall quantile [0.7,0.9]	0.29	0.24	0.23	0.53	0.29	0.23	0.57	0.25	0.25	0.57	0.31	0.28	0.53	0.28	0.24
Recall quantile [0.9,1.0]	0.20	0.30	0.18	0.27	0.29	0.16	0.32	0.23	0.22	0.32	0.38	0.22	0.29	0.24	0.16
F-Measure quantile [0,0.1]	0.09	0.14	0.13	0.13	0.25	0.15	0.08	0.17	0.18	0.08	0.17	0.17	0.14	0.20	0.17
F-Measure quantile [0.1,0.3]	0.19	0.25	0.21	0.23	0.24	0.24	0.22	0.18	0.26	0.22	0.19	0.20	0.25	0.24	0.21
F-Measure quantile [0.3,0.7]	0.47	0.46	0.53	0.55	0.54	0.48	0.57	0.54	0.52	0.57	0.57	0.55	0.56	0.53	0.48
F-Measure quantile [0.7,0.9]	0.28	0.26	0.13	0.34	0.22	0.20	0.39	0.17	0.07	0.39	0.08	0.16	0.38	0.22	0.19
F-Measure quantile [0.9,1.0]	0.07	0.13	0.13	0.11	0.20	0.13	0.09	0.13	0.13	0.09	0.16	0.15	0.13	0.17	0.16
Cohen's Kappa	0.0495	0.0559	0.0496	0.1409	0.1111	0.0523	0.1620	0.0594	0.0617	0.1620	0.0783	0.0749	0.1644	0.0916	0.0567

Exhibit 8: Model performance metrics for horizons $H_{\Delta\tau}$ computed on the test folds. The column labels H5, H60, H300 refer to $H_{\Delta\tau}|\ \Delta\tau = 5$, $H_{\Delta\tau}|\ \Delta\tau = 60$, $H_{\Delta\tau}|\ \Delta\tau = 300$, respectively.

Overall, LSTM and CNN-LSTM models outperformed MLP and CNN models, but the performance gap is not big. All these four deep learning models performed significantly better than the simple linear Logistic Regression model, indicating there is a strong non-linear relationship worth exploring. We have highlighted the best performing model for each performance statistics. Note that the Averaged Accuracies (around 0.30) are much lower than the category-size Weighted Accuracies (around 0.7), due to drastic accuracy difference among different categories (quantiles) -- the middle not-much-change quantile [0.3, 0.7] enjoys high accuracy rates of 0.80, while the big-drop quantile [0, 0.1] and the big-jump quantile [0.9, 1.0] barely reach accuracy of 0.16, only slightly better than a random model of 0.1 accuracy (the size of the extreme quantile). This supports the practical observation that it is much harder to predict large market movements than stable market fluctuations.

The sub-par precision for predicting large market movements make model-driven trading strategy unprofitable. The models need to be further improved with more stocks, with longer time frame, and with better architecture, before model-driven trading strategy can generate any profits.

5. Conclusions

This paper presents a benchmark LOB dataset of China stock market, covering a few thousand stocks for the period of June to September 2020. Experiment protocols are designed for model performance evaluation: at the end of every second, to forecast the upcoming volume-weighted average price (VWAP) change and volume over 12 horizons ranging from 1 second to 300 seconds. Results based on linear and state-of-the-art deep learning models (CNN, LSTM, CNN-LSTM) are compared. Practical short-term trading strategy framework based on the alpha signal generated is presented. The data and code are available on Github (https://github.com/HKGSAS).

The first-phase experiment results based on only 20 actively traded stocks have shown the

promise of deep learning models outperform linear model by a large margin. However, the accuracy level for predicting large market movements is marginally better than a random model. Further research can be conducted to include more data, longer history, and improved deep learning architecture. We welcome researchers around the world to cooperate and to further develop the field of applying advanced deep learning models to analyze financial time series data and to explore profitable trading opportunities.

Acknowledgment

The research has received feedback and comments from [].

References

[1] F. Abergel, M. Anane, A. Chakraborti, A. Jedidi, and I. Toke. Limit Order Books. PHYSICS OF SOCIETY: ECONOPHYSICS. Cambridge University Press, 2016.

[2] A. Briola, J. Turiel and T. Aste. Deep Learning modeling of Limit Order Book: a comparative perspective. ArXiv abs/2007.07319, 2020.

[3] M. D. Gould, M. A. Porter, S. Williams, M. McDonald, D. J. Fenn, and S. D. Howison, Limit order books, Quantitative Finance, vol. 13, no. 11, pp. 1709–1742, 2013.

[4] R. Huang and T. Polak. Lobster: Limit order book reconstruction system. Available at SSRN 1977207, 2011.

[5] A. N. Kercheval and Y. Zhang. Modelling high-frequency limit order book dynamics with support vector machines. Quantitative Finance, 15(8):1315–1329, 2015.

[6] A. Ntakaris, M. Magris, J. Kanniainen, M. Gabbouj, and A. Iosifidis, Benchmark dataset for mid-price forecasting of limit order book data with machine learning methods, Journal of Forecasting, vol. 37, no. 8, pp. 852– 866, 2018.

[7] J. Sirignano and R. Cont. Universal features of price formation in financial markets: perspectives from deep learning. Quantitative Finance, 19(9):1449–1459, 2019.

[8] Z. Zhang, S. Zohren, and S. Roberts. DeepLOB: Deep convolutional neural networks for limit order books. IEEE Transactions on Signal Processing, 67(11):3001–3012, June 2019.