


# Deep Learning Option Price Movement

Weiguan Wang <sup>\*,†</sup>  and Jia Xu <sup>†</sup>

School of Economics, Shanghai University, 333 Nanchen Road, Baoshan District, Shanghai 200444, China

\* Correspondence: weiguanwang@shu.edu.cn

† These authors contributed equally to this work.

**Abstract:** Understanding how price-volume information determines future price movement is important for market makers who frequently place orders on both buy and sell sides, and for traders to split meta-orders to reduce price impact. Given the complex non-linear nature of the problem, we consider the prediction of the movement direction of the mid-price on an option order book, using machine learning tools. The applicability of such tools on the options market is currently missing. On an intraday tick-level dataset of options on an exchange traded fund from the Chinese market, we apply a variety of machine learning methods, including decision tree, random forest, logistic regression, and long short-term memory neural network. As machine learning models become more complex, they can extract deeper hidden relationship from input features, which classic market microstructure models struggle to deal with. We discover that the price movement is predictable, deep neural networks with time-lagged features perform better than all other simpler models, and this ability is universal and shared across assets. Using an interpretable model-agnostic tool, we find that the first two levels of features are the most important for prediction. The findings of this article encourage researchers as well as practitioners to explore more sophisticated models and use more relevant features.

**Keywords:** limit order book; model agnostic; neural networks; options

**JEL Classification:** G13; G120



**Citation:** Wang, Weiguan, and Jia Xu. 2024. Deep Learning Option Price Movement. *Risks* 12: 93. <https://doi.org/10.3390/risks12060093>

Academic Editor: Angelos Dassios

Received: 13 May 2024

Revised: 29 May 2024

Accepted: 30 May 2024

Published: 4 June 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Much of the market microstructure research is about understanding how trading affects price and liquidity dynamics (Easley et al. 2021). Such understandings are also meaningful for market makers who regularly place limit orders based on their prediction of critical microstructure characteristics to make profit and reduce market risk. The continuing adoption of electronic trading and its faster frequency have accumulated a massive amount of data, for which simple market microstructure models may not be applicable. By studying a tick-level intraday data, this article intends to shed light on how prices are impacted by real-time trading activities in a high-frequency trading environment.

To this end, we demonstrate in this article how machine learning can provide a different viewpoint on the market microstructure for option markets. Machine learning has shown great success in many prediction tasks since it can extract a non-linear and complex relationship given a large amount of data. In this article, we investigate the predictability of option price movement based on limit order book (LOB). Options are financial derivatives that offer the holder the right to buy/sell an asset at a certain strike price sometime in the future. Hence, in contrast to equities, they have a few characteristics that makes their price movement prediction more difficult. First, the prices of different options are more closely related since they are written on the same underlying asset (for index options). This could provide an additional source of predictability. Second, options are much less traded than equities, hence having a different market microstructure. Simply put, it is worthwhile investigating the applicability of machine learning on this asset class.

In this paper, we take the prices and volumes of the five levels of buy and sell orders in the limit order book as predictive variables. The dataset being studied comprises tick-level intraday data, as they reveal subtle market behaviors within shorter periods that are often not observable in traditional daily data. The directions of price movement of corresponding options are the dependent variable. Thus, a three-class classification problem is investigated.

This formulation has been adopted frequently in the literature. [Ntakaris et al. \(2018\)](#) offer the first publicly available benchmark dataset of high-frequency limit order markets for mid-price prediction. They extract normalized data representations of time series data for five stocks from the Nasdaq stock market for 10 consecutive days. This dataset is used later in a few studies, such as that of [Zhang et al. \(2019\)](#). [Easley et al. \(2021\)](#) construct several classic market microstructure measures and show that they continue to provide insights into the price process in current complex markets. They only use random forests to handle classification problems of changes in various market microstructure indicators on futures data, including changes in bid–ask spread, volatility, and liquidity. They find that microstructure features with high explanatory power do not always lead to high predictive power. Instead of constructing a classification problem, [Kolm et al. \(2023\)](#) use deep learning in forecasting high-frequency returns at multiple horizons for 115 stocks traded on Nasdaq using order book information at the most granular level. They basically predict a return term structure using a long short-term memory neural network and hand-crafted features called order-imbalance flows. Their results show that “information-rich” stocks can be predicted more accurately.

This article contributes to the rapidly growing literature on applying deep learning models to financial data ([Gu et al. 2020](#); [Liu et al. 2022](#)). The novelty of this paper lies in three aspects. First, previous papers mostly focus on applying machine learning to primary asset limit order book, such as equities (see [Kolm et al. \(2023\)](#); [Sirignano and Cont \(2019\)](#); [Zhang et al. \(2019\)](#)) or foreign exchanges (see [Ito et al. \(2022\)](#)). However, the predictability of a derivative limit option book has not been well studied yet. The outcome of this study encourages researchers to build more informative features to improve the performance of prediction. Second, we confirm that complex models such as deep neural networks are able to predict the mid-price movement better than simple models, also highlighted by [Kelly et al. \(2023\)](#) in the case of equity market timing. Third, we apply an interpretability analysis on the importance of features to provide a machine learning viewpoint on the market microstructure. Overall, this work is an empirical exploration of the challenges brought about by high-frequency trading and the application of machine learning.

Our main findings are as follows. First, complex deep learning models predict the price trends of options more accurately than simpler models. A three-layer long short-term memory (LSTM) model with multiple time-lagged features achieves the best performance among all candidate models. Second, we discover that the universality of the predictive model exists, i.e., a model train on one asset can be deployed to predict on another asset in the spirit of transfer learning. This property allows one to train a complex model for a new product for which the data are not rich yet. Last, we apply a local interpretable model-agnostic tool to provide an explanation for the prediction of a deep neural network model. It turns out that the first two levels of features are more important than the rest. This observation aligns with common knowledge and enhances our confidence in deep learning models. A trading simulation shows that predictions based on deep learning models result in economic gains with a minor drawdown.

### 1.1. Literature Review

Predicting asset price movement has been an important part of market microstructure research. This literature review introduces related works in two categories, price-volume (parametric) models and machine learning models.

### 1.1.1. Price-Volume (Parametric) Models

Market microstructure models trade off between execution price and waiting cost using limit order. On the theoretical side, O'hara (1998) is a classic introductory textbook. Goettler et al. (2005) model a dynamic limit order market as a stochastic sequential game with rational traders. They find transaction costs paid by market order submitters are negative on average, and are negatively correlated with spread. Roşu (2009) propose a model of an order-driven market where fully strategic, symmetrically informed liquidity traders dynamically choose between limit and market orders. It suggests that high competition leads to smaller spread and lower price impact, buy and sell orders can cluster away from bid–ask spread, and the bid and ask price shows comovement.

On the empirical side, there is ample evidence in the literature that price movement is predictable with price-volume information, particularly order imbalance. For instance, Chordia et al. (2002) investigate the role of the aggregate daily order imbalance, buy order less sell orders. They find that it increases following market declines and vice versa. Order imbalance in either direction reduces liquidity. In addition, market returns are strongly affected by contemporaneous and lagged order imbalances. Chordia and Subrahmanyam (2004) further investigate the relation between order imbalance and daily return of stocks. They find that price pressures caused by autocorrelated imbalances lead to a positive relation between returns and lagged imbalances. However, this relation reverses sign after controlling for the current imbalance.

Traditionally, researchers use price-volume information at a daily resolution (Jiang et al. 2020). As the high-frequency limit order book data have become more readily available and of higher quality, they are now an important source of information. Harris and Panchapagesan (2005) investigate the informativeness of the limit order book regarding future price changes and whether specialists utilize this information in their trading. They conclude that the limit order book provides insights into future price movements. According to Cao et al. (2009), about 78% of the information is from the best bid and ask price, while the other higher levels offer the remaining 22%. According to Cont et al. (2014), the influence of order book events on price changes over short time frames is primarily due to the order flow imbalance (OFI), which is the disparity between the supply and demand at the best bid and ask prices. Their results reveal a linear relationship between OFI and price changes, with the slope being inversely related to market depth. Rather than focusing on the price-volume relationship, Chan and Fong (2000) explore how the number of trades, trade sizes, and order imbalance (initiated by buyers versus sellers) contribute to explaining the volatility–volume relationship in a sample of NYSE and Nasdaq stocks.

### 1.1.2. Machine Learning Models

High-frequency limit order book datasets are inherently of a vast volume, allowing for the deployment of machine learning and deep learning models which are successful in discovering hidden relationship in the context of big data. Compared to the linear feature such as order flow imbalance, this non-parametric approach of modeling can learn a non-linear representation. Recently, the most widely used models include the long short-term memory neural network and decision tree. Some researchers have been using raw order book features for predicting price movement, see Sirignano and Cont (2019); Zhang et al. (2019), while others advocate using hand-crafted features that are proven to be useful in the literature (Easley et al. 2021; Kolm et al. 2023). The former allows various machine learning models to extract representation that has never been discovered before, while the latter essentially imposes regularization based on domain knowledge.

Multiple types of machine learning models have been applied to order book prediction problems. Easley et al. (2021) use random forest. Sirignano (2019); Sirignano and Cont (2019) apply feed-forward neural networks and compare with the vector auto-regression model. Tashiro et al. (2019) propose to utilize market order together with limit order, and use the convolutional neural network to predict stock price trend. Zhang et al. (2019) propose to combine convolutional and long short-term memory neural networks, also used

by Tsantekidis et al. (2020). Sidogi et al. (2023) propose to use a signature transformation for dimension reduction and combine with several machine learning methods. To facilitate the training and increase information content, Ntakaris et al. (2019, 2020) conduct large-scale feature engineering to include more than 270 technical features for the stock trend prediction task. Huang et al. (2021) kindly provide a benchmark order book dataset for a few thousand Chinese stocks, and apply neural network models on it. Other related works can be found in Arroyo et al. (2024); Lucchese et al. (2024); Zhang et al. (2021).

The advancement of machine learning allows to study cross-asset impact in a non-parametric way. Cross-asset impact refers to the fact that price impact of trading on one asset can affect another. Easley et al. (2021) point out that “There is no market microstructure theory of how these cross-asset effects should, or even could, occur, and there are many plausible alternatives”. As an initial exploration, they include in the feature set for the classification problem not only the own microstructure features of an asset but also a shared subset of assets chosen to represent actively traded futures. Their findings indicate that a small number of cross-market features are beneficial for the prediction of multiple output variables, and asset own features are not always the most important ones. Sirignano and Cont (2019) take a data enlargement approach, with aggregate samples from multiple assets, and show that the performance of machine learning models improve. Hence, the relationship between price change and predictive characteristics learned from one asset can also be applied to another asset.

The rest of the paper is organized as follows: Section 1.1 provides a brief literature review, Section 2 describes the dataset and cleaning steps. Section 3 introduces various models used for the forecasting task. Section 4 presents the results. Section 5 concludes.

## 2. Data and Variables

### 2.1. Data and Preprocessing

The data used in this study consist of a high-frequency options limit order book obtained from the Wind Database. The dataset includes two types of equity exchange traded fund (ETF) options; they are options on 50 ETF (code: 510050) and on 300 ETF (Shanghai Stock Exchange, code: 510300), respectively<sup>1</sup>. In the Chinese market, they are among the most liquid equity options. These two option limit order book data provide a valuable source for understanding market price formation. Our data cover a time period of 24 months, from January 2020 to December 2021. Each sample in the dataset represents an instantaneous snapshot of multiple levels of limit order book information of an option. Figure 1 illustrates such a sample. It shows the bid price, bid size, ask price, and ask size of five levels, resulting in a total of 20 features.

	Size	Price
Asks	10	¥0.4309
	10	¥0.4235
	10	¥0.4216
	20	¥0.4086
	20	¥0.4082
Bids	20	¥0.4010
	20	¥0.4006
	10	¥0.3902
	1	¥0.3877
	10	¥0.3849

**Figure 1.** A sample of the order book. The ‘Asks’ side represents sell orders, while the ‘Bids’ side represents buy orders. ‘Size’ indicates the number of options available for sale/purchase at a given price, and ‘Price’ represents the order’s buying or selling price. The best bid and ask refer to the highest bid and lowest ask prices, respectively.

To enhance the quality of the high-frequency order book data, we implement several filters. First, we remove samples that have a best ask price lower than the best bid price,

which are caused by instantaneous market fluctuation. Additionally, we eliminate samples with a trading volume of zero (Cao and Han 2013), which are typically generated by changes at deeper levels of the order book. These changes do not correspond to actual buy and sell orders and thus have a relatively mild impact on the mid-price movement. Including these samples may lead to model instability. Lastly, we exclude data for options that have not yet expired by the end of the time period covered by our dataset. In the end, we retain approximately 150 million samples for subsequent analysis and modeling.

Improper partitioning of the dataset may lead to information leakage issues, especially by violating the time series structure; see an example in Wang and Ruf (2022). We divide the dataset into three non-overlapping time periods to maintain the temporal order of the data, with training, validation, and test sets spanning 16, 4, and 4 months. The sizes of the datasets are shown in Table 1. The model is first trained on the training set to learn the mapping from the feature of the data to the target value. During training, the validation set is used to assess the model's performance and determine the optimal parameters. The validation set also helps prevent the model from overfitting to the training data, thus enhancing its generalization ability. The test set is used for the final evaluation of the model's performance.

**Table 1.** Dataset partition and size.

Set	Time Length	Time Range	No. of Options	Sample Size (10 K)
Training	16	2020.01–2021.04	2096	11,533
Validation	4	2021.05–2021.8	572	2906
Test	4	2021.9–2021.12	379	1994

Table 2 lists the data descriptive statistics, after cleaning. Since the entire dataset is too large, with about 150 samples, only a random batch of data samples (32,768 samples) is selected here for brevity. The overall data distribution is similar. The dataset has twenty variables. It is convenient to include lagged features in deep learning models, hence the total number of features for such models may add up to hundreds.

**Table 2.** Descriptive statistics of variables. There are twenty variables in total, being five levels of ask price, bid price, ask size and bid size. Here, only a random batch of samples (32,768 samples) is used to calculate descriptive statistics. The overall distribution is similar.

	Mean	Std	Min	25%	50%	75%	Max
askprice1	0.099	0.109	0.000	0.027	0.068	0.134	1.426
askprice2	0.099	0.109	0.000	0.027	0.068	0.134	1.428
askprice3	0.100	0.109	0.000	0.028	0.068	0.134	1.430
askprice4	0.100	0.109	0.000	0.028	0.068	0.134	1.433
askprice5	0.100	0.109	0.000	0.028	0.069	0.135	1.434
bidprice1	0.099	0.108	0.000	0.027	0.068	0.133	1.424
bidprice2	0.098	0.108	0.000	0.027	0.068	0.133	1.423
bidprice3	0.098	0.108	0.000	0.027	0.067	0.133	1.421
bidprice4	0.098	0.108	0.000	0.027	0.067	0.133	1.420
bidprice5	0.098	0.107	0.000	0.027	0.067	0.132	1.418
asksize1	31.222	75.964	1.000	9.000	15.000	32.000	3674.000
asksize2	37.760	69.594	0.000	10.000	20.000	43.000	2816.000
asksize3	37.631	67.262	0.000	10.000	20.000	41.000	2421.000
asksize4	35.498	68.580	0.000	10.000	20.000	40.000	3920.000
asksize5	33.144	63.252	0.000	10.000	20.000	37.000	4521.000
bidsize1	32.778	89.341	0.000	9.000	15.000	34.000	4885.000
bidsize2	41.880	110.408	0.000	10.000	20.000	45.000	10,580.000
bidsize3	41.233	103.241	0.000	10.000	20.000	43.000	9136.000
bidsize4	38.526	85.076	0.000	10.000	20.000	40.000	4577.000
bidsize5	36.176	73.618	0.000	10.000	20.000	39.000	2747.000

## 2.2. Variable

We consider the information from the five levels of the order book, comprising a total of 20 features, as the state of the order book. Define the state of the order book of an option at time  $t$  as

$$\text{LOB}_t = \left( p_1^a(t), v_1^a(t), p_1^b(t), v_1^b(t), \dots, p_5^a(t), v_5^a(t), p_5^b(t), v_5^b(t) \right) \in \mathbb{R}^{20},$$

where  $p_i^a$ ,  $p_i^b$ ,  $v_i^a$ , and  $v_i^b$  represents the ask price, bid price, ask size, and bid size at level  $i$  of the order book, respectively. To take into account the temporal correlation in asset price movements, we construct lagged time series for each sample. The lag order  $p$  is a hyper-parameter to be tuned. Hence, each sample  $X_i$  is given by

$$X_i = (\text{LOB}_{t-p+1}, \text{LOB}_{t-p+2}, \dots, \text{LOB}_t).$$

This time series is the actual input feature of each sample  $i$ .

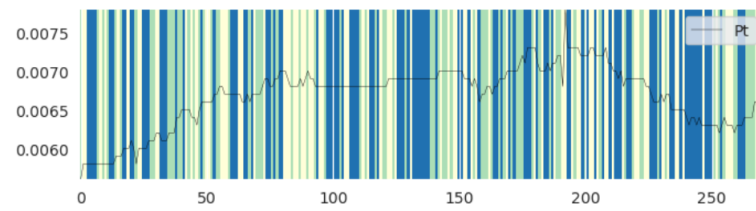
## 2.3. Label Construction

The mid price is given by the average of the best bid price and the best ask price,  $m_t = \frac{1}{2}(p_1^a(t) + p_1^b(t))$ . Our target variable is the direction of the next movement of the mid-price. There exist various ways of defining such a variable in the literature. By doing so, one actually introduces two hyper-parameters that require tuning which could make the performance dependent on these quantities, sometimes to a large extent. To avoid introducing extra hyper-parameters, we adopt a straight-forward definition in this paper. Define the difference between mid-prices of two consecutive orders as  $d_{t+1} := m_{t+1} - m_t$ . When  $d_{t+1} > 0$ , we classify it as an uptrend in the next moment's mid-price. When  $d_{t+1} = 0$ , the mid-price remains unchanged, and when  $d_{t+1} < 0$ , it is a downtrend. When constructing labels, we convert these three categories into one-hot coding as the outputs.

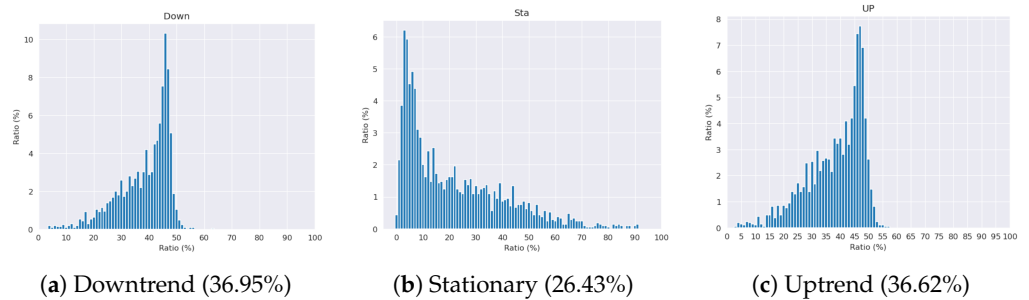
In some works such as [Sirignano and Cont \(2019\)](#), researchers eliminate samples where the mid-price remains unchanged when cleaning data. This approach simplifies the prediction problem from a three-class to a binary classification problem. The fact that the majority of changes in the high-frequency order book appear in the higher levels and thus do not lead to actual mid-price changes makes the movement direction highly imbalanced. It is difficult for deep learning models to perform well in such an imbalanced dataset. Removing the unchanging mid-price part can effectively avoid this issue. However, this approach has a critical drawback: the prediction is thus conditional on a change happening. In other words, this model is only applicable when the agent knows how to predict that there will be a change in the subsequent orders. [Bouchaud et al. \(2018\)](#) also point out that changes in the order books' shape also affect prices in addition to actual trades. Indeed, we will see that our dataset is well balanced in terms of the three labels, ruling out the necessity of removing stationary samples.

To obtain a better feeling of the dataset, [Figure 2](#) shows the price trend of an option (code '1000191') on 5 January 2020. We see that the direction of the price movement changes frequently, and the duration of each direction also varies. The update time distances of option order book information are rather irregular, from a fraction of a second to several seconds.

[Figure 3](#) shows histograms of proportions for each of the three label categories across all option contracts. The left graph represents the distribution for downtrends, the middle for unchanged prices, and the right for uptrends. We calculate the overall number of downward, stationary, and upward movements as percentages of all samples, and show these quantities in the bottom of each panel of [Figure 3](#). The downtrends and uptrends account for 39.95% and 36.62% of the total sample, while the stationary samples account for 26.43%. According to the above statistics and analysis, there is no evident imbalance in the data.



**Figure 2.** Option price trend and direction of change chart. This chart displays the price trend of the option with the code '10001919' on 5 January 2020. The  $x$ -axis represents the sequence of trades for the option throughout the day, and the  $y$ -axis indicates the price. The curve represents the actual price path of the option. The blue shaded area in the graph indicates that the price remains unchanged in the next moment, light blue signifies a price increase, and yellow represents a price decrease.



**Figure 3.** Histogram of the proportions of the three types of labels. The panel (a) graph accounts for downtrends, (b) for being stationary, and (c) for uptrends. In each graph, the  $x$ -axis represents the proportion of the corresponding labels for each option relative to the total number of samples for that option. The  $y$ -axis then shows the percentage of such options relative to the total number of options. For instance, if 3% of options have 40% of their next moment's price movements as downtrends, then this is reflected by the bar (40%, 3%) on the left panel.

### 3. Models and Evaluation Metrics

In this section, we introduce various machine learning and deep learning models that will be used later for the prediction task. They include decision tree, random forest, logistic regression and long-short term neural networks.

#### 3.1. Long Short-Term Memory Model

As the implementation of this model is well established and the relevant literature is readily accessible, we will only provide a brief description here; for a more detailed introduction, readers can refer to Goodfellow et al. (2016). LSTM models are composed of numerous units, each consisting of a cell, an input gate, an output gate, and a forget gate. The expressions for these components are given by the following formulas:

$$\begin{aligned}
 \mathbf{f}_t &= \text{sigmoid}(\mathbf{W}_f \mathbf{x}_t + \mathbf{U}_f \mathbf{h}_{t-1} + \mathbf{b}_f), \\
 \mathbf{i}_t &= \text{sigmoid}(\mathbf{W}_i \mathbf{x}_t + \mathbf{U}_i \mathbf{h}_{t-1} + \mathbf{b}_i), \\
 \mathbf{o}_t &= \text{sigmoid}(\mathbf{W}_o \mathbf{x}_t + \mathbf{U}_o \mathbf{h}_{t-1} + \mathbf{b}_o), \\
 \mathbf{c}_t &= \mathbf{f}_t \circ \mathbf{c}_{t-1} + \mathbf{i}_t \circ \tanh(\mathbf{W}_c \mathbf{x}_t + \mathbf{U}_c \mathbf{h}_{t-1} + \mathbf{b}_c), \\
 \mathbf{h}_t &= \mathbf{o}_t \circ \tanh(\mathbf{c}_t).
 \end{aligned}$$

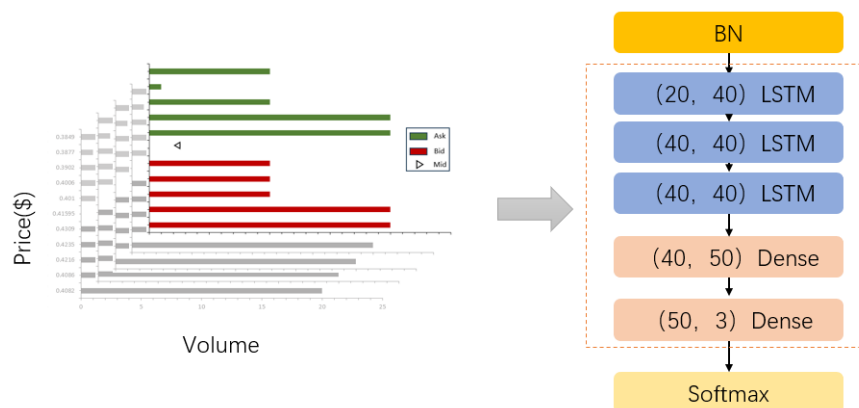
In these formulas,  $\mathbf{x}_t$  represents the input vector to the unit,  $\mathbf{h}_t$  the cell state,  $\mathbf{f}_t$  the activation vector of the forget gate,  $\mathbf{i}_t$  the activation vector of the input gate,  $\mathbf{o}_t$  the activation vector of the output gate, and  $\mathbf{c}_t$  the hidden state vector. The activation functions for the input, output, and forget gates are sigmoid function, which monotonically map inputs to interval (0, 1). The symbol  $\circ$  denotes element-wise multiplication. Information in the LSTM model is stored in the cell state  $\mathbf{h}$  and hidden state  $\mathbf{c}$ , where the former represents short-term memory and the latter long-term memory in the model.

Figure 4 shows the input data and LSTM structure. The left side of the figure illustrates the input data, where each sample is composed of the current state of the order book and multiple past states. The right panel shows the LSTM structure. Our network consists of three LSTM layers with 40 nodes, and two fully connected layers with 50 nodes. Before entering the LSTM layers, we apply batch normalization to adjust the mean and variance of each batch, a technique that accelerates neural network training and enhances model generalization.

We use cross-entropy as the loss function and Adam optimizer to train the neural network. To improve the performance, we tune hyper-parameters based on the model’s performance on the validation set. The range and optimal values of hyper-parameters are listed in Table 3. The optimal values are chosen such that deviating from these values on either side leads to worse performance on the validation set. Due to the computational constraints, we choose a relatively larger batch size to speed up model training. The model in this paper is implemented using Pytorch and runs on an NVIDIA RTX4090 GPU. Each training of LSTM with twenty lagged features takes roughly 48 h.

**Table 3.** Hyper-parameters of LSTM.

Parameter	Lag Order	Learning Rate	Batch Size	Epoch	Hidden States
Optimal value	50	$10^{-3}$	$2^{16}$	56	40
Tuning range	0–50	$10^{-2}$ – $10^{-4}$	$2^8$ – $2^{16}$	50–60	20–50



**Figure 4.** Input data and LSTM neural network structure. The (left) panel displays the state of the limit order book at a specific moment and its preceding 10 moments. Each graph’s horizontal axis represents the volume of orders waiting to be traded, and the vertical axis shows different levels of buy and sell prices. The (right) panel shows the LSTM structure used in our experiments. BN represents the batch normalization layer, and the three LSTM labels indicate the three layers of LSTM. The number 20 indicates that the input data comprise twenty features, and 40 represents the size of the hidden states in the LSTM model. The dense layer is a fully connected layer. The output activation function is softmax.

### 3.2. Benchmark Models

In this section, we introduce benchmark models to provide a baseline for predicting mid-price movements. These models are typically simpler than the stacked LSTM model, with fewer parameters and less complexity. They also fit more rapidly, without the need for lengthy gradient descent methods, parallel computing frameworks, or specialized hardware such as GPUs. The benchmark models used in this paper include the decision tree, random forest, and multiclass logistic regression. All the three models can be implemented with the scikit-learn package in Python.



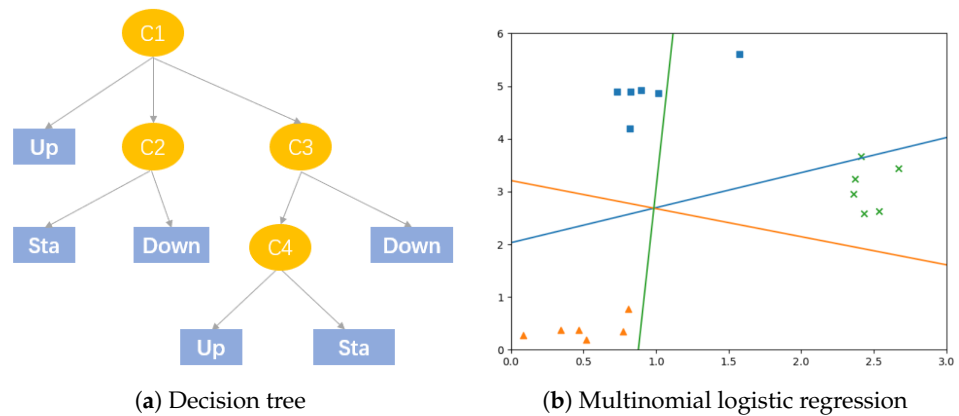
### 3.2.1. Decision Tree and Random Forest

Decision tree models have more straightforward decision rules and are thus considered ‘white-box’. Figure 5a shows a schematic plot of decision tree on a three-class classification problem. A decision tree recursively partitions the input space into regions based on feature values and fits a simple model on each region. A partition corresponds to a decision node. In a node  $m$ , representing a region  $R_m$  with  $N_m$  observations, let

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i = k),$$

be the proportion of class  $k$  observations in node  $m$ . Then, the observations are classified to the majority class. Classification trees are optimized by minimizing the Gini index. For more details on decision trees, we refer to [Hastie et al. \(2008\)](#).

Decision tree are prone to overfitting. Their size depends on the preset tree depth. Excessive depth can lead to overly fine segmentation of the observation space, resulting in poor performance on test data. One regularization approach is to train multiple decision trees to form a random forest. In the prediction phase, the majority voting method is used to determine the forest’s output label, i.e., the most common label among the base decision trees. In this paper, we set the number of trees to be 100.



**Figure 5.** Decision tree and logistic regression. The panel (a) panel shows the structure of a simple three-level decision tree model. Yellow circles represent internal nodes, i.e., splitting rules. Blue rectangles are leaf nodes, representing model outcomes. The panel (b) shows the classification results of the multinomial logistic regression model. Each colour of the markers represents a category. Each line corresponds to a decision boundary.

### 3.2.2. Multinomial Logistic Regression

The third model is multinomial logistic regression. It calculates the probability of each input sample belonging to each category using a linear function and softmax function, then selects the category with the highest probability as the prediction result.

The model has the form

$$\log \frac{P(y = k|x)}{P(y = K|x)} = \beta_k^T x$$

for each class  $k = 1, \dots, K$ . The model is specified in terms of  $K - 1$  log-odds, hence summing up to one. Logistic regression models are usually fit by maximum likelihood. Again, we refer to [Hastie et al. \(2008\)](#) for an introduction. Figure 5b shows a schematic plot of the logistic regression.

### 3.3. Evaluation Metrics

The most commonly used criterion of model selection for a classification task is accuracy. It measures the ratio of samples whose predicted labels match the true ones, expressed as

$$\sum_{i=1}^N \mathbf{1}(\hat{y}_i = y_i) / N,$$

where  $y_i$  and  $\hat{y}_i$  represent the true and predicted labels of the  $i$ -th sample, respectively,  $N$  denotes the total number of samples, and  $\mathbf{1}$  denotes the indicator function. [Easley et al. \(2021\)](#) have used this criterion to select a model as well as to determine feature importance. The higher the accuracy, the better the model identifies correct labels.

While accuracy is the criterion for model selection, we also report precision, recall, and the  $F1$  score to better understand performance. Let TP denotes true positive samples, FP false positive, FN false negative, and TN true negative. Then, precision is defined as the percentage of true positive among all predicted positive samples, and recall is the percentage of true positive among all actual positive samples:

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad \text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}.$$

The  $F1$  score can be interpreted as a harmonic mean of the precision and recall, given by

$$F1 = \frac{2 * \text{TP}}{2 * \text{TP} + \text{FP} + \text{FN}}.$$

Although these three metrics were originally designed for binary classification problems, we can independently calculate these metrics for each label category and then use the macro average to compute the three final metrics. We shall see below that these measures choose the same model as the accuracy criterion. Last, we evaluate using the confusion matrix to reveal the performance for each individual class. For an introduction to these metrics, refer to [Goodfellow et al. \(2016\)](#).

## 4. Results

In this section, we present our empirical results in three parts. Section [4.1](#) discusses the various model performance on a dataset that includes both types of options. Section [4.2](#) investigates the universal ability of predictive models that are trained on one asset but applied on another asset. Using a linear approximation method, Section [4.3](#) intends to unveil the predictive power of deep neural networks by investigating the importance of features and their financial meaning.

### 4.1. Baseline Experiment Results

In the first experiment, we use the entire dataset that includes both the options on 50 ETF and on 300 ETF. It is split into training, validation, and test sets with the lengths of each being 16, 4, and 4 months. The sizes of each dataset are given in [Table 1](#).

We compare several benchmark models that include the decision tree, random forest, multinomial logistic regression, against the LSTM models. Since the benchmark models cannot inherently deal with the time series structure, we do not feed time-lagged features for them. The LSTM model instead is intentionally designed to utilize long-term information; hence, we compare the same LSTM model with lagged features of different steps. To fairly compare benchmark models with the LSTM, we also evaluate the LSTM(1) model that is trained without lagged features.

[Table 4](#) shows the out-of-sample comparison results of the LSTM model with 1, 5, 10, 20, and 50 time steps against other methods with one time step. We roughly arrange the models in ascending order of performance. Since the three types of labels are roughly evenly distributed, a simplistic model would achieve 36% accuracy by predicting all samples to be upward, according to the label distribution in [Figure 3](#). Hence, all the models tested

here have exceeded this simplistic approach and demonstrated their abilities to predict the movement direction of the option limit order book to different extents. The LSTM models have slightly better results with increasing time order of the samples. We can see that the LSTM(50) model outperforms all other models in terms of all metrics. Figure 6a displays the LSTM(50) loss on the training and validation sets as we train for 60 epochs, and Figure 6b shows the accuracy on these sets. As the training reaches the end, the model loss stabilizes and the accuracy peaks, leading us to retain this optimal model. Since the training of LSTM (50) takes about 48 h, and the performance in terms of accuracy is not much higher than LSTM(20), we decide to choose the latter for efficiency. Among the three benchmark models, the multinomial logistic regression achieves 51.59% accuracy, about two percent worse than the LSTM(1) model. This shows that the non-linear transformation in the LSTM cell helps improve the prediction ability.

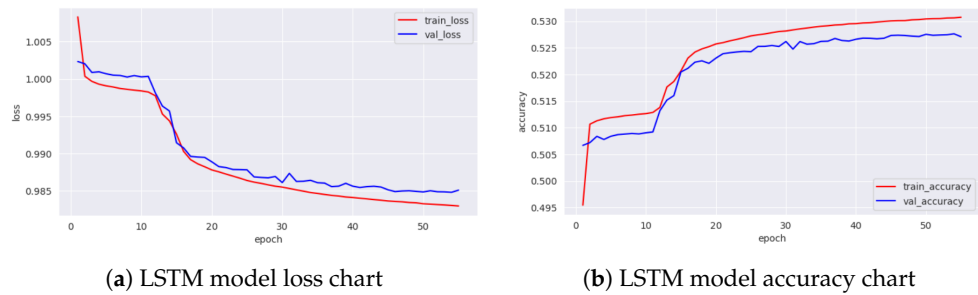


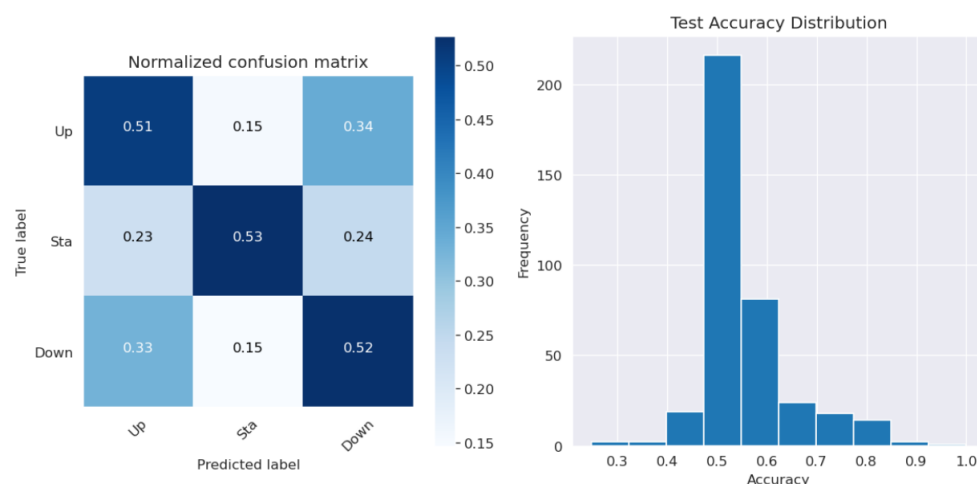
Figure 6. Loss function and accuracy.

Table 4. Predictive performance of various models on out-of-sample (test set) for all options. The decision tree, random forest, logistic regression, and LSTM(1) models have used one time step feature to predict the incoming moment, while LSTM(5), LSTM(10), LSTM(20) and LSTM(50) use the past five, ten, twenty, and fifty lagged time steps.

Model	Accuracy	Precision	Recall	F1
Decision tree	42.38%	42.41%	42.38%	42.35%
Random forest	50.53%	50.90%	50.53%	50.61%
Logistic regression	51.59%	52.02%	51.59%	51.68%
LSTM(1)	53.36%	51.12%	50.73%	50.75%
LSTM(5)	53.26%	51.17%	50.77%	50.80%
LSTM(10)	53.31%	51.07%	50.75%	50.79%
LSTM(20)	53.45%	50.97%	50.70%	50.68%
LSTM(50)	53.47%	52.55%	51.91%	52.07%

To further explore our conclusions, we present the confusion matrix of the prediction of the LSTM(50) model on all samples of the test set in the left panel of Figure 7. From the figure, it is evident that 50% or more of the actual price movement directions were correctly predicted.

Apart from the overall performance, we further investigate whether the optimal model can predict well for each option in the test set, shown in the right panel of Figure 7. We can see that the accuracy distribution has a mean well above 50%. Even in the worst case, the model can still predict about 40% for an individual option, indicating that the model performs consistently across options in the test data without extreme high or low outliers. Hence, we have confidence that the LSTM (50) model does not predict awfully in extreme market scenarios.



**Figure 7.** LSTM(50) model confusion matrix (left) and accuracy distribution of individual options (right). For the left panel, the vertical axis represent the actual direction distribution of price changes, and the horizontal axis shows the predicted direction by the model. The right panel plots the histogram of the prediction accuracy of the model on each individual option.

#### 4.2. Cross-Asset Performance

Deep learning models tend to perform better when more data are collected for training. This inspires researchers to look for a universal property, which means that a predictive model trained on one asset can be applied on another asset. For instance, [Sirignano and Cont \(2019\)](#) find that a model trained on a large pool of stocks can outperform stock-specific models with high confidence, even when the pool of stocks exhibit great heterogeneity. If such universality exists, one can first train a predictive model on a large number of assets for which data are rich, and then apply to a specific asset for which the data are scarce. This kind of transfer learning methodology could be useful for markets in which new stocks or assets are issued often.

To test this universality, we divide the option data based on different underlying assets. We use the 50 ETF options for training and validation, and the 300 ETF options for out-of-sample prediction. Again, we split according to the time series order for different underlying assets. The summary of partition is given in Table 5.

**Table 5.** Dataset division for options with different underlyings.

Set	Underlying Asset	Time Range	Number of Options	Samples (10 K)
Training	50 ETF	2020.01–2021.04	752	5464
Validation	50 ETF	2021.05–2021.08	214	1423
Test	300 ETF	2021.09–2021.12	92	781

The model performance is shown in Table 6. We ignore the other LSTM models except the LSTM(20) for simplicity as well as efficiency. We can see that the LSTM(20) model still performs the best on the test set, with an accuracy of 52.72%. Note that all models perform slightly worse than using all data. For the LSTM(20), its decrease is about one percent, similar to the decision tree and random forest. The multiclass logistic regression model has the smallest decrease in accuracy. An explanation for this is that the logistic regression model is the simplest among all, thus showing less variance. We can conclude that there exists a universality in the option limit order book, providing evidence for promoting the transfer learning methodology.

**Table 6.** Cross-asset predictive performance. See the caption of Table 4 for an explanation. We ignore the LSTM model of other number of lagged features except 20, for the sake of simplicity.

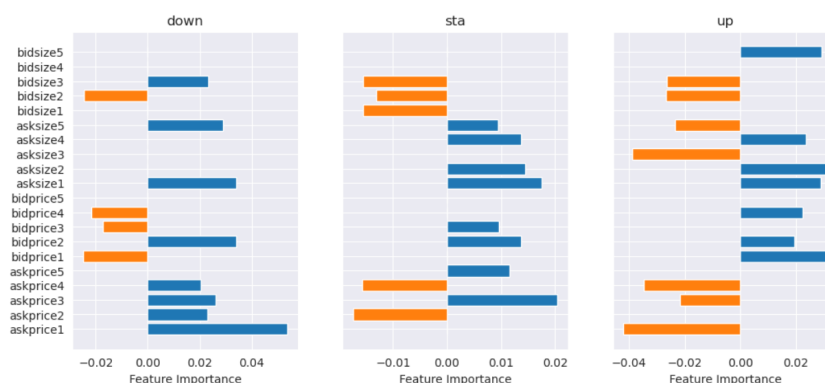
Metric	Accuracy	Precision	Recall	F1
Decision tree	41.78%	41.81%	41.78%	41.75%
Random forest	49.65%	49.91%	49.65%	49.63%
Logistic regression	51.11%	51.32%	51.11%	51.12%
LSTM(20)	<b>52.72%</b>	<b>51.43%</b>	<b>51.28%</b>	<b>51.33%</b>

### 4.3. Feature Importance

So far, we have compared the deep neural networks with white-box models such as decision trees in terms of performance, and show that the more complicated models are advantageous. However, this comparison does not investigate the complicated models themselves, i.e., we do not know how the complicated model makes the decision. If we can understand the relationship between the input components and model predictions, we can compare these relationships with our domain knowledge to decide whether we can accept or reject the predictions. This insight is also important in finance research.

To calculate the importance of features, we apply the LIME (Local Interpretable Model-agnostic Explanations<sup>2</sup>) proposed by Ribeiro et al. (2016) to explain complex models that are typically hard to interpret, such as deep neural networks. The implementation process is as follows. First, we select a sample and generate a set of “similar” data points around this sample. Then, we feed these newly generated data points into the original model to observe how the model predictions change with slight variations in the input. Next, LIME fits a simple, locally approximate, and highly interpretable model, such as linear regressions, using these similar samples and their corresponding outputs from the original model. Finally, this simple model reveals the importance of different features of the sample to the prediction.

Figure 8 shows the importance of twenty features for a sample, displaying how the LSTM reacts to the given inputs. In the figure, a blue bar indicates a positive influence of the feature on the corresponding class, while a orange bar means a negative effect. The length of the bars represents the magnitude of influence. One can see that the low-level features, such as the first and second level sizes and prices, have more influence on the prediction, while the rest are minor. This observation aligns well with our common understanding on the market microstructure, making the predictions more credible. Although we have only shown the explanation for a particular sample, this observation holds as more samples are tested, which is not presented here for brevity.



**Figure 8.** The importance of each feature for the LSTM model. The feature importance is calculated for each label class independently, hence there are three panels. The blue bars mean that the features have positive effects on the corresponding label, and the orange bars mean negative effects. The predicted label is stationary.

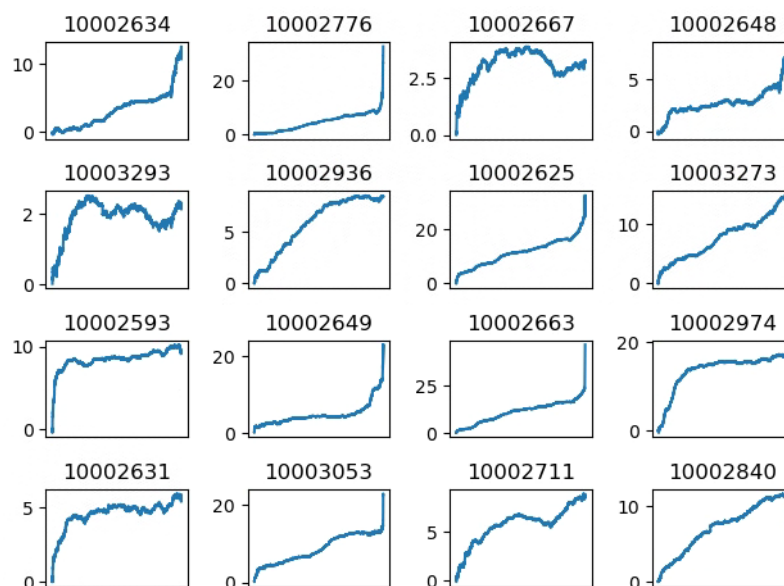
#### 4.4. A Trading Simulation

Having demonstrated the predictive performance of various models, we aim to test their practicability in terms of economic terms. In this subsection, we use a trading strategy based on predictions of the best performing model, the LSTM model. The number of feature time lags is 50.

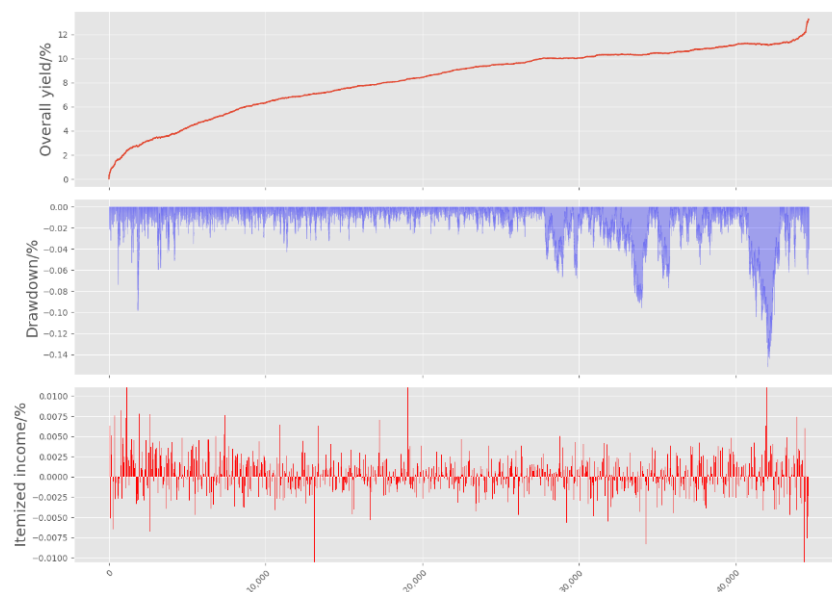
We set the number of options being traded to one. In practice, one can optimize this quantity according to market conditions, including liquidity, limit order book depth, and so on. This further improves the performance of any trading strategy. For simplicity, we choose the unit size so that our strategy is not affected by the market impact issue.

Our strategy is as follows. We buy one unit of options when the model predicts that the mid-price will increase. The option is sold when the model predicts the downtrend. Otherwise, no action is taken. Following Zhang et al. (2019), we assume mid-trade, where mid-prices are the execution price. This corresponds to a trading style where one enters the trade passively and exits aggressively.

Figure 9 displays the cumulative returns of trading sixteen options individually, according to our previously designed strategy. All the plots show positive returns. Among them, some return curves increase steadily, while others exhibit fluctuations. Most profit curves are not completely linear, with some experiencing accelerated profit growth or slowdowns at specific times. To see the overall performance, Figure 10 presents the average return, maximal drawdown, and individual profit per trade in the three panels, respectively. The top panel suggests that the average profit is smoothly growing with little fluctuation. The middle graph depicts the maximal drawdown rate of the strategy, with a maximum rate as little as  $-0.14\%$ . The bottom graph shows the return rate for each transaction. It indicates that positive returns, both in frequency and magnitude, far exceed the negative ones.



**Figure 9.** This chart presents the cumulative returns of sixteen options traded according to our designed strategy. Each sub-figure’s horizontal axis represents time, and the vertical axis represents the return rate, with the title of each graph being the option code traded. For simplicity, we multiplied all return rates by 100, meaning a value of 20 in the chart corresponds to an actual return rate of 20%.



**Figure 10.** Overall cumulative return (**top**), maximal drawdown (**middle**), and individual return (**bottom**).

## 5. Conclusions and Discussion

We investigate the ability of recurrent neural networks in option price movement prediction, and compare it with that of other classification models, leading to the following conclusions. First, deep neural networks are superior in predictive ability compared to other supervised machine learning methods and traditional linear classifiers. Second, there exists a universal relationship in price formation common to all options. Third, we confirm that the low-level features contribute more to determining the movement direction. Our major contribution is discovering the predictability of the short-term price trend in a derivative market, and the advantage of complex models and more features.

Given different market characteristics, this article concurs with much of the existing literature in that the superior performance of the deep learning model in predicting short-term price trends still holds in the Chinese option market, similar to [Sirignano and Cont \(2019\)](#) for the stock market and [Easley et al. \(2021\)](#) for the future market. The fact that the performance gradually improves as more lagged features are used in neural network models also confirms the observations in [Cont et al. \(2023\)](#) and [Nian et al. \(2021\)](#). It actually leads one to consider using dimension reduction approaches on the input feature space for better representation to further improve prediction.

Our findings are also meaningful for practitioners, especially market makers who regularly place orders based on their predictions. First, this article further encourages them to deploy deep learning methods in trading. Second, the cross-asset predictability provide a way for them to build trading strategies when new products are issued to the market and few data are available. Last, due to the fact that performance improves as more data are used, it encourages practitioners to use a larger amount and a greater variety of data.

This paper has a few limitations, which are left for future research. First, although our experimental results show that deep learning algorithms perform well in predicting option prices, to determine whether this predictability can form a truly profitable trading strategy, more considerations should be taken, such as transaction fees, commissions, margins, etc. To deal with these, one should consider a better label construction, or optimizing transaction cost during model training. Second, one may consider including hand-crafted features found meaningful in the existing literature and testing their importance.

**Author Contributions:** W.W.: Conceptualization, Methodology, Validation, Formal analysis, Data curation, Resources, Writing—Review and Editing, Supervision, Funding acquisition, Project administration. J.X.: Software, Formal analysis, Investigation, Writing—Original Draft, Visualization. All authors have read and agreed to the published version of the manuscript.

**Funding:** Wang thanks the support by the National Natural Science Foundation of China, grant number: 72201158.

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author.

**Acknowledgments:** Wang gives thanks for the advice from Yupeng Jiang.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Notes

- <sup>1</sup> Both ETFs are intended to track two widely-used stock indices in China. SSE 50 Index is the stock index of Shanghai Stock Exchange, representing the top 50 companies by “float-adjusted” capitalization. The CSI 300 is a broader capitalization-weighted stock market index designed to replicate the performance of the top 300 stocks traded on the Shanghai Stock Exchange and the Shenzhen Stock Exchange. The former has a much higher trading volume than the latter. In this paper, we only use the 300 ETF options from the Shanghai Stock Exchange.
- <sup>2</sup> The authors have made the tool available as a python package at <https://github.com/marcotcr/lime> (accessed on 1 December 2023).

## References

- Arroyo, Alvaro, Alvaro Cartea, Fernando Moreno-Pino, and Stefan Zohren. 2024. Deep attentive survival analysis in limit order books: Estimating fill probabilities with convolutional-transformers. *Quantitative Finance* 24: 35–57. [CrossRef]
- Bouchaud, Jean-Philippe, Julius Bonart, Jonathan Donier, and Martin Gould. 2018. *Trades, Quotes and Prices: Financial Markets under the Microscope*. Cambridge: Cambridge University Press.
- Cao, Charles, Oliver Hansch, and Xiaoxin Wang. 2009. The information content of an open limit-order book. *Journal of Futures Markets* 29: 16–41. [CrossRef]
- Cao, Jie, and Bing Han. 2013. Cross section of option returns and idiosyncratic stock volatility. *Journal of Financial Economics* 108: 231–49. [CrossRef]
- Chan, Kalok, and Wai-Ming Fong. 2000. Trade size, order imbalance, and the volatility–volume relation. *Journal of Financial Economics* 57: 247–73. [CrossRef]
- Chordia, Tarun, and Avanidhar Subrahmanyam. 2004. Order imbalance and individual stock returns: Theory and evidence. *Journal of Financial Economics* 72: 485–518. [CrossRef]
- Chordia, Tarun, Richard Roll, and Avanidhar Subrahmanyam. 2002. Order imbalance, liquidity, and market returns. *Journal of Financial Economics* 65: 111–30. [CrossRef]
- Cont, Rama, Arseniy Kukanov, and Sasha Stoikov. 2014. The price impact of order book events. *Journal of Financial Econometrics* 12: 47–88. [CrossRef]
- Cont, Rama, Mihai Cucuringu, and Chao Zhang. 2023. Cross-impact of order flow imbalance in equity markets. *Quantitative Finance* 23: 1373–93. [CrossRef]
- Easley, David, Marcos López de Prado, Maureen O’Hara, and Zhibai Zhang. 2021. Microstructure in the machine age. *The Review of Financial Studies* 34: 3316–63. [CrossRef]
- Goettler, Ronald L., Christine A. Parlour, and Uday Rajan. 2005. Equilibrium in a dynamic limit order market. *The Journal of Finance* 60: 2149–92. [CrossRef]
- Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. Cambridge: MIT Press.
- Gu, Shihao, Bryan Kelly, and Dacheng Xiu. 2020. Empirical asset pricing via machine learning. *The Review of Financial Studies* 33: 2223–73. [CrossRef]
- Harris, Lawrence E., and Venkatesh Panchapagesan. 2005. The information content of the limit order book: Evidence from nyse specialist trading decisions. *Journal of Financial Markets* 8: 25–67. [CrossRef]
- Hastie, Trevor, Robert Tibshirani, and Friedman Jerome. 2008. *The Elements of Statistical Learning Data Mining, Inference, and Prediction*. Berlin and Heidelberg: Springer.
- Huang, Charles, Weifeng Ge, Hongsong Chou, and Xin Du. 2021. Benchmark dataset for short-term market prediction of limit order book in china markets. *The Journal of Financial Data Science* 3: 171–83. [CrossRef]
- Ito, Katsuki, Hitoshi Iima, and Yoshihiro Kitamura. 2022. LSTM forecasting foreign exchange rates using limit order book. *Finance Research Letters* 47: 102517. [CrossRef]



- Jiang, Minqi, Jiapeng Liu, Lu Zhang, and Chunyu Liu. 2020. An improved Stacking framework for stock index prediction by leveraging tree-based ensemble models and deep learning algorithms. *Physica A: Statistical Mechanics and its Applications* 541: 122272. [\[CrossRef\]](#)
- Kelly, Bryan T., Semyon Malamud, and Kangying Zhou. 2023. The virtue of complexity in return prediction. *The Journal of Finance* 79: 459–503. [\[CrossRef\]](#)
- Kolm, Petter N., Jeremy Turiel, and Nicholas Westray. 2023. Deep order flow imbalance: Extracting alpha at multiple horizons from the limit order book. *Mathematical Finance* 33: 1044–81. [\[CrossRef\]](#)
- Liu, Qingfu, Zhenyi Tao, Yuman Tse, and Chuanjie Wang. 2022. Stock market prediction with deep learning: The case of China. *Finance Research Letters* 46: 102209. [\[CrossRef\]](#)
- Lucchese, Lorenzo, Mikko S. Pakkanen, and Almut E. D. Veraart. 2024. The short-term predictability of returns in order book markets: A deep learning perspective. *International Journal of Forecasting*, in press. [\[CrossRef\]](#)
- Nian, Ke, Thomas F. Coleman, and Yuying Li. 2021. Learning sequential option hedging models from market data. *Journal of Banking & Finance* 133: 106277.
- Ntakaris, Adamantios, Giorgio Mirone, Juho Kannianen, Moncef Gabbouj, and Alexandros Iosifidis. 2019. Feature engineering for mid-price prediction with deep learning. *IEEE Access* 7: 82390–412. [\[CrossRef\]](#)
- Ntakaris, Adamantios, Juho Kannianen, Moncef Gabbouj, and Alexandros Iosifidis. 2020. Mid-price prediction based on machine learning methods with technical and quantitative indicators. *PLoS ONE* 15: e0234107. [\[CrossRef\]](#)
- Ntakaris, Adamantios, Martin Magris, Juho Kannianen, Moncef Gabbouj, and Alexandros Iosifidis. 2018. Benchmark dataset for mid-price forecasting of limit order book data with machine learning methods. *Journal of Forecasting* 37: 852–66. [\[CrossRef\]](#)
- O'hara, Maureen. 1998. *Market Microstructure Theory*. Hoboken: John Wiley & Sons.
- Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. 2016. "Why should i trust you?" explaining the predictions of any classifier. Paper presented at the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13–17, pp. 1135–44.
- Roşu, Ioanid. 2009. A dynamic model of the limit order book. *The Review of Financial Studies* 22: 4601–41. [\[CrossRef\]](#)
- Sidogi, Thendo, Wilson Tsakane Mongwe, Rendani Mbuvha, Peter Olukanmi, and Tshilidzi Marwala. 2023. A signature transform of limit order book data for stock price prediction. *IEEE Access* 11: 70598–609. [\[CrossRef\]](#)
- Sirignano, Justin A. 2019. Deep learning for limit order books. *Quantitative Finance* 19: 549–70. [\[CrossRef\]](#)
- Sirignano, Justin, and Rama Cont. 2019. Universal features of price formation in financial markets: Perspectives from deep learning. *Quantitative Finance* 19: 1449–59. [\[CrossRef\]](#)
- Tashiro, Daigo, Hiroyasu Matsushima, Kiyoshi Izumi, and Hiroki Sakaji. 2019. Encoding of high-frequency order information and prediction of short-term stock price by deep learning. *Quantitative Finance* 19: 1499–506. [\[CrossRef\]](#)
- Tsantekidis, Avraam, Nikolaos Passalis, Anastasios Tefas, Juho Kannianen, Moncef Gabbouj, and Alexandros Iosifidis. 2020. Using deep learning for price prediction by exploiting stationary limit order book features. *Applied Soft Computing* 93: 106401. [\[CrossRef\]](#)
- Wang, Weiguan, and Johannes Ruf. 2022. A note on spurious model selection. *Quantitative Finance* 22: 1797–800. [\[CrossRef\]](#)
- Zhang, Zihao, Bryan Lim, and Stefan Zohren. 2021. Deep learning for market by order data. *Applied Mathematical Finance* 28: 79–95. [\[CrossRef\]](#)
- Zhang, Zihao, Stefan Zohren, and Stephen Roberts. 2019. Deeplob: Deep convolutional neural networks for limit order books. *IEEE Transactions on Signal Processing* 67: 3001–12. [\[CrossRef\]](#)

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.