UNIVERSITY OF EDINBURGH
Business School

2022-23

CMSE11528 FINANCIAL MACHINE LEARNING II

HFT Limit Order Book Forecasting

B213323

Word counts:

1,998

Introduction

This report examines the performance of various machine learning algorithms, including LSTM, LSTM Autoencoders, MLP, XGBoost, and Random Forests, in predicting mid-price movements in Limit Order Book (LOB) data for a regression task. Among these algorithms, LSTM demonstrates the best performance. Incorporating eleven new features designed to capture the complex dynamics of the LOB enhances the prediction outcomes for both XGBoost and LSTM models. Furthermore, this study compares the extracted features from LSTM Autoencoders with 11 manually crafted features, finding that the handcrafted features yield superior results in prediction effectiveness.

Literature Review

The literature on LOB data prediction is vast and diverse, with various machine learning algorithms and feature engineering techniques being proposed and evaluated. A key challenge in this domain is the temporal dependencies in LOB data, which must be considered to achieve accurate predictions. Ntakaris et al. (2019) found that Long Short-Term Memory (LSTM) networks, a type of recurrent neural network, are well-suited to capturing these temporal dependencies and, thus, perform well in predicting mid-price movements.

Another critical aspect of LOB prediction is feature engineering, which involves selecting and creating relevant features from the raw data to improve the model's performance. In their study, Ntakaris et al. (2019) compared manual feature crafting with fully automated feature extraction using LSTM Autoencoders and found that the manual approach outperforms the automated one. This suggests that handcrafted features can better capture the underlying dynamics of the LOB and contribute to more accurate predictions.

Ensemble methods, such as Random Forests and XGBoost, have gained popularity in the machine learning community due to their ability to improve prediction accuracy and reduce overfitting. These algorithms typically rely on decision trees and have shown promising results in various applications, including financial time series forecasting ((Caruana and Niculescu-Mizil, 2006); (Chen and Guestrin, 2016)).

Data Pre-processing

The target variable for the regression task, the mid-price, was first examined. It was observed that 2,932 rows contained zero values for mid-price labels. Further analysis revealed that either AskPrice1 or BidPrice1 was missing, or in some cases, both were absent. Additionally, numerous rows contained zero values, which could adversely affect the regression task, causing the model to learn irrelevant information from the label. These missing labels are likely due to periods of low trading activity when no other data is available at subsequent levels. Since the missing labels account for just 0.52% of the total rows, it was deemed appropriate to remove these rows to clean the data.

Subsequently, the data was divided into 60% for training, 20% for validation, and 20% for testing. It was essential to reformat the data to accommodate the requirements of LSTM and LSTM AE models. This is because LSTM algorithms consider a specific length of data that needs to be retained within the model's memory.

Feature Engineering and Feature Reduction

Enhancing the model's performance is possible by increasing relevant features and decreasing irrelevant ones. Ntakaris et al. (2019) found that a manual feature crafting approach outperforms a fully automated feature extraction method using LSTM Autoencoders. To improve upon the initial set of 40 features, which primarily represent the state of the LOB, an additional 11 features have been incorporated into the dataset as follows:

1. Financial Duration

   This feature measures the time difference between consecutive data points, which reflects the market activity level (Ntakaris *et al.*, 2019a). By including this feature, we account for variations in the trading frequency, capturing time-varying liquidity and potential changes in market participants' behaviour.

2. MidPrice_OIB

   The order imbalance weighted mid-price (MidPrice_OIB) is an alternative to the regular mid-price (Ntakaris *et al.*, 2019a). It weighs the bid and asks prices by their respective volumes, emphasising the price levels where most trading interest is concentrated. This can provide insights into the perceived value of the asset by market participants.

3. VolImbalance

   The volume imbalance represents the proportion of bid volume to the total volume at the best bid and asks prices (Ntakaris *et al.*, 2019a). This feature captures the immediate supply and demand imbalance, which may impact short-term price movements.

4. BA_spread

   The bid-ask spread represents the difference between the best bid and asks prices (Ntakaris *et al.*, 2019a). It is an essential measure of liquidity and can also reflect the uncertainty and risk associated with trading the asset.

5. AccumulatedVolumeDifference

   This feature captures the LOB's supply and demand imbalance by calculating the difference between the accumulated bid and asks volumes across multiple levels (Cont and Kukanov, 2014). It reflects the market's inclination towards buying or selling the asset soon.

6. WeightedAverageBidPrice

   The weighted average bid price, calculated across multiple bid levels, provides an aggregated view of the price levels market participants are willing to buy (Cartea and Jaimungal, 2015).

7. WeightedAverageAskPrice

   Similarly, the weighted average asks price aggregates the price levels market participants are willing to sell (Cartea and Jaimungal, 2015).

8. OrderFlowImbalance

   This feature measures the LOB's short-term buying or selling pressure by calculating the difference between the bid and asks volumes across multiple levels. It can offer insights into the short-term momentum of the market (Biais, Hillion and Spatt, 1995).

9. RelativeBidVolume

   This feature represents the current best bid volume ratio to the average bid volume across multiple levels (Ranaldo, 2004). It captures the relative liquidity at the best bid price.

10. RelativeAskVolume

    The relative ask volume measures the current best ask volume ratio to the average ask volume across multiple levels, reflecting the liquidity at the best ask price (Ranaldo, 2004).

11. PriceMomentum

    The price momentum feature compares the current mid-price with the mid-price of a specified number of periods ago. It captures the price trends and helps to identify potential continuation or reversal patterns in the market (Jegadeesh and Titman, 1993).

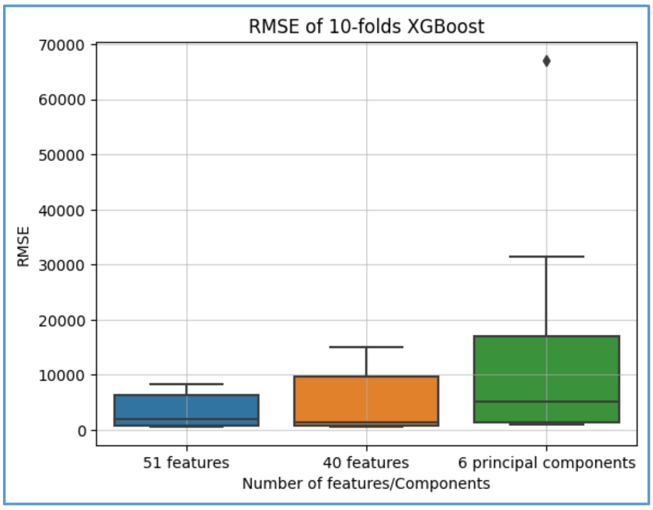

Figure 1 RMSE of XGBoost for different features.

Figure 1 presents the outcomes of incorporating the novel features into the XGBoost regressor model. When comparing the mean RMSE calculated using 10-fold time series cross-validation, the XGBoost model with the additional features outperforms the model with only 40 features. Interestingly, extracting six principal components, which account for 99% of the variance, did not improve the mean RMSE for the XGBoost model.

Model Selection, Model Rejection and Result

In this experimental protocol, seven different regression algorithms were evaluated and compared for predicting mid-price movement, as follows:

1. LSTM trained with 40 original features and 20 latent features extracted from LSTM Autoencoder.
2. LSTM trained with 20 latent features extracted from LSTM Autoencoder.
3. MLP trained with 40 original features.
4. Vanilla LSTM trained with 40 original features.
5. Random Forests trained with 40 original features.
6. XGBoost trained with 40 original features.
7. Vanilla LSTM trained with 40 original features and additional 11 hand-crafted features.

The seven algorithms employed in this study were subjected to online learning iterations using the same dataset, partitioned into smaller segments. In this approach, after each learning iteration, the model was saved and subsequently loaded for the next iteration. This method enabled the models to continuously learn and adapt to new data as it became available, effectively simulating a real-world scenario where the models must incorporate newly acquired information in their predictions.

A closer examination of Table 1 reveals that the Vanilla LSTM algorithm with 40 original features performed the best among the six models, with the lowest median RMSE of 5368.760173. This result indicates that the Vanilla LSTM algorithm was able to capture the complex dynamics of the LOB and make accurate predictions of the mid-price movements. The success of the Vanilla LSTM model can be attributed to its ability to learn and retain information from long sequences of data, which is essential for predicting the mid-price movements in LOB data (Ntakaris et al., 2019).

Table 1 RMSE results of online learning from six algorithms.

Online iteration	LSTM AE 60 features	LSTM AE 20 latent	MLP	Vanilla LSTM	Random Forests	XGBoost
1	6801.212989	6669.309465	6405.397087	6111.198211	7370.568034	7092.782294
2	6796.652324	6725.706775	6464.807039	5466.380136	6922.046901	5733.497829
3	75220.25376	80807.9289	85305.36442	40798.79827	89262.53276	88679.67588
4	31062.86466	6362.48498	36112.70157	27032.98565	31297.25676	4920.329302
5	6116.52472	5307.332408	6359.069232	5271.140211	6399.077296	6182.607953
6	1453.591891	1708.29498	1389.73472	583.5261255	318.946214	275.3053548
7	4490.239894	5149.406566	2445.770109	2000.321467	4629.383784	2656.590514
8	2116.571344	2019.415155	2135.162202	1983.669234	2452.387788	2612.019615
9	1973.031398	1926.461759	2000.930815	1828.362925	1997.751038	2065.600576
10	8469.755017	12184.12209	6396.830073	12153.31278	13174.06132	9475.655667
Mean	14450.0698	12886.04631	15501.57673	10322.9695	16382.40119	12969.4065
Median	6456.588522	5834.908694	6377.949653	5368.760173	6660.562098	5326.913565

Interestingly, adding 20 latent features extracted from LSTM Autoencoder to the LSTM model (LSTM AE 60 features) did not improve its performance. The model with 60 features performed worse than the Vanilla LSTM model with 40 original features, with a median RMSE of 6456.588522. This finding aligns with the observation by Ntakaris et al. (2019) that a manual feature crafting approach outperforms a fully automated feature extraction method using LSTM Autoencoders. The additional latent features extracted from the LSTM Autoencoder may have needed to have been more relevant and informative to improve the model's performance in predicting the mid-price movements.

The MLP algorithm, a simple feedforward neural network, also demonstrated relatively good performance with a median RMSE of 6377.949653. However, it still fell short of the performance achieved by the Vanilla LSTM model. This result suggests that the temporal dependencies in the LOB data, which are effectively captured by the LSTM algorithm, play a crucial role in predicting mid-price movements.

Despite their reputation as state-of-the-art ensemble methods in machine learning, the Random Forests and XGBoost algorithms demonstrated relatively poor performance compared to the other models. This result can be attributed to the fact that these algorithms, based on decision trees, are not inherently designed to capture the temporal dependencies in time series data, such as LOB data, which can be crucial for making accurate predictions (Ntakaris et al., 2019).

The LSTM model was then modified by adding 11 manually crafted features obtained from feature engineering to evaluate if these additional features enhance the performance in terms of RMSE. The outcome of this experiment is presented in Table 2.

Table 2 The RMSE comparison between two identical LSTM models but trained with different features. The first model was trained with 40 features, while the second was trained with additional 11 handcrafted features.

Online iteration	Vanilla LSTM 40 features	Vanilla LSTM 51 features
1	6111.198211	6289.685838
2	5466.380136	5412.297879
3	40798.79827	49965.17651
4	27032.98565	11721.47102
5	5271.140211	5391.029203
6	583.5261255	499.2290114
7	2000.321467	2607.214134
8	1983.669234	2067.023139
9	1828.362925	1687.047611
10	12153.31278	14648.22632
Mean	10322.9695	10028.84007
Median	5368.760173	5401.663541

Table 2 shows that the LSTM model trained with 51 features, including the 11 handcrafted features, has a slightly better mean RMSE value (10028.84007) than the model trained with only 40 features (10322.9695). The median RMSE values are also quite similar, with the 51-feature model having a marginally lower value (5401.663541) than the 40-feature model (5368.760173). These results suggest that including the 11 handcrafted features provides a modest improvement in the prediction performance of the LSTM model. The additional features capture more complex aspects of the Limit Order Book data, contributing to better predictive accuracy in mid-price movements.

Conclusion

This report investigated the performance of various machine learning algorithms, including LSTM, LSTM Autoencoders, MLP, XGBoost, and Random Forests, in predicting mid-price movements in Limit Order Book (LOB) data for a regression task. The Vanilla LSTM algorithm with 40 original features demonstrated the best performance, emphasising the importance of capturing temporal dependencies in the data for accurate predictions. Furthermore, the addition of 11 handcrafted features provided a modest improvement in the prediction performance of the LSTM model, supporting the findings of Ntakaris et al. (2019) that manual feature crafting can outperform automated feature extraction using LSTM Autoencoders.

However, it is essential to acknowledge the limitations of this study, such as the potential for overfitting due to the relatively small sample size and the assumption of a stationary LOB market, which may only sometimes hold. Future research should explore the performance of these algorithms in different market conditions, as well as the inclusion of more sophisticated features or alternative methods to capture the complex dynamics of LOB data.

References

Biais, B., Hillion, P. and Spatt, C. (1995) ‘An Empirical Analysis of the Limit Order Book and the Order Flow in the Paris Bourse’.

Cartea, Á. and Jaimungal, S. (2015) ‘RISK METRICS AND FINE TUNING OF HIGH-FREQUENCY TRADING STRATEGIES: risk matrics and hf trading strategies’, *Mathematical Finance*, 25(3), pp. 576–611. Available at: https://doi.org/10.1111/mafi.12023.

Caruana, R. and Niculescu-Mizil, A. (2006) ‘An empirical comparison of supervised learning algorithms’, in *Proceedings of the 23rd international conference on Machine learning - ICML ’06*. *the 23rd international conference*, Pittsburgh, Pennsylvania: ACM Press, pp. 161–168. Available at: https://doi.org/10.1145/1143844.1143865.

Cont, R. and Kukanov, A. (2014) ‘Optimal order placement in limit order markets’. arXiv. Available at: http://arxiv.org/abs/1210.1625 (Accessed: 12 April 2023).

Jegadeesh, N. and Titman, S. (1993) ‘Returns to Buying Winners and Selling Losers: Implications for Stock Market Efficiency’.

Ntakaris, A. *et al.* (2019) ‘Feature Engineering for Mid-Price Prediction With Deep Learning’, *IEEE Access*, 7, pp. 82390–82412. Available at: https://doi.org/10.1109/ACCESS.2019.2924353.

Ranaldo, A. (2004) ‘Order aggressiveness in limit order book markets’, *Journal of Financial Markets*, 7(1), pp. 53–74. Available at: https://doi.org/10.1016/S1386-4181(02)00069-1.

Bibliography

Christopher M Bishop (1995) *Neural Networks for Pattern Recognition*. Clarendon Press.

Goodfellow, I., Bengio, Y. and Courville, A. (2016) *Deep learning*. Cambridge, Massachusetts: The MIT Press (Adaptive computation and machine learning).

Keras (2023) 'Keras official documentation'. Available at: https://keras.io/getting_started/.

Makridakis, S. *et al.* (2022) 'Statistical, machine learning and deep learning forecasting methods: Comparisons and ways forward', *Journal of the Operational Research Society*, pp. 1–20. Available at: https://doi.org/10.1080/01605682.2022.2118629.

Michael Nielsen (2015) *Neural Networks and Deep Learning*. Determination Press. Available at: http://neuralnetworksanddeeplearning.com/.

Ntakaris, A. *et al.* (2018) 'Benchmark dataset for mid-price forecasting of limit order book data with machine learning methods', *Journal of Forecasting*, 37(8), pp. 852–866. Available at: https://doi.org/10.1002/for.2543.

scikit-learn (2023) 'Scikit-learn official documentation'. Available at: https://scikit-learn.org/stable/.

Tsantekidis, A. *et al.* (2020) 'Using Deep Learning for price prediction by exploiting stationary limit order book features', *Applied Soft Computing*, 93, p. 106401. Available at: https://doi.org/10.1016/j.asoc.2020.106401.

XGboost (2023) 'XGBoost official documentation'. Available at: https://xgboost.readthedocs.io/en/stable/.