Financial Time-series Analysis for High-Frequency Trading

Kumsetty Nikhil Venkat - 181IT224
Dept. of Information Technology
NITK Surathkal
Mangaluru, India
nikhilvenkat26@gmail.com

Amith Bhat - 181IT105
Dept. of Information Technology
NITK Surathkal
Mangaluru, India
amithbhat01@gmail.com

Arya Sharma - 181IT207
Dept. of Information Technology
NITK Surathkal
Mangaluru, India
arya.181it207@nitk.edu.in

***Abstract*—Predicting financial succession has long been a challenging issue due to the noisy and volatile nature of the market. In the field of High-Frequency Trading (HFT), predicting trading objectives is a very challenging task because the automated process of requiring information requires precision and speed. In this project, we have used an unconventional construction method for predicting time series data, which provides the results of modern technology. These technologies are trained and tested in the prescribed bookmark Limit Order Book (LOB) FI-2010, and the corresponding results are compared and analyzed using a variety of methods.**

***Index Terms*—Feed-forward neural network, Bilinear projection, Temporal Attention, Financial data analysis, Time-series prediction**

I. INTRODUCTION

Time series divisions and predictions have been widely studied in various domains. Examples include natural language analysis [1], medical data analysis, finance and economics and the general phases of time .The complex dynamic of financial, visual data is stagnant and noisy, representing a limited view of the basic creative process. This makes financial speculation in the timeline one of the most difficult tasks in the series of time series [2].

The rise of specialized hardware and software has enabled a comprehensive collection of business information, which is also an opportunity and challenge for retailers, especially the most advanced retailers. In addition to long-term investments, HFT is characterized by high speed and short-term investment time. The ability to analyze large amount of data in a short period of time is essential to HFT.

Over the decades, several types of statistics have been proposed to exclude financial features from sound, time-consuming financial systems, such as the Auto-Regressive Integrated Moving Average (ARIMA) [3], [4] incorporating variability to eliminate ambiguity. However, to ensure exposure, these types are often made under the concepts of many basic data-sharing concepts, leading to misalignment in future observations. Therefore, machine learning models such as Logistic Regression and Support Vector Machines were used, which give better results than ARIMA in various situations.

Although the above-mentioned machine learning models perform well, they are not specifically designed to integrate temporary data into time series data. The neural network design phase called Recurrent Neural Networks (RNN) is specially designed to extract temporary data from consecutive green data. RNNs began to gain popularity in many different application areas [5], [1], [6] recently due to improved computer and computation hardware, as well as the availability of more information. Deep neural networks work directly on the introduction of raw data instead of hand-made objects. As a result, relevant data-dependent features are automatically removed, improving the performance and robustness of the entire system.

While deep neural networks are generally and LSTM networks, in particular, are biologically inspired and efficient at work, trained structures are often difficult to interpret. Also, while not focusing on architecture that often improves performance and comprehension, it also includes higher computer costs across the model. This precludes the implementation of the model in many financial forecasting scenarios where the ability to quickly train the system and make predictions with large degrees of continuous input data plays an important role. Therefore, RNNs are not ready for financial forecasts.

Therefore, in this project, we have implemented a new type of multivariate time-series data layer construction. The proposed structure is designed to use the concept of bilinear speculation by introducing attention to the temporary mode. The LOB bench database, the FI-2010 database, was used as a database.

II. Literature Survey

A. Related Work

In this project, we have taken [20] as a basic paper - using and developing the model proposed by its authors.

Deep neural networks are artistic entities not only in the functions of human comprehension, such as language and image comprehension but also in predicting complex time series data. For example, RNN networks based on LSTM constructions have been used to predict future rainfall intensity in a variety of geographical areas, material use [17], or to detect patterns in a series of clinical periods.

In analyzing financial data, Deep Belief Networks and AutoEncoders were used to obtain trading portfolio models [11], [12]. Besides, Deep Reinforcement Certified Learning Methods are also popular among the financial asset trading models category [13], [14]. The spatial correlation between LOB levels was studied [15] by a perceptron (MLP) consisting of 3 hidden fragments measuring the distribution of bid-sharing and query values.

Due to the volatility, sounds of stock price movements, many deep neural networks are proposed within a complex predictive pipeline. By making the LOB status normal with pre-feeding date statistics on CNN [9] or the LSTM network [10]. A transparent pipeline containing multiresolution wavelet transform to filter sound input series, Auto-Encoder stacked to produce a high-quality representation of each stock index, an LSTM network to predict future pricing is recently proposed in [15]. Finally, the authors [21] propose a DeepLOB model that uses multiple CNN layers before wrapping them up with two LSTM layers for enhanced reading.

B. Motivation

Most of the in-depth learning models in the High-Frequency Trading (HFT) market currently contain special unfamiliar special models such as SVM, CNN, or duplicate structures such as LSTM. Some custom DL models are too deep or less efficient, both of which have serious disadvantages in the highly competitive HFT sector.

C. Problem Statement

Creating a custom-designed deep learning model with two hidden layers to distinguish whether stock prices will rise, fall, or remain stagnant than other predictions. This is done by using the concept of bilinear speculation and by inserting the focus method into a temporary mode to make it more efficient.

D. Objectives

- To create a model with simpler structures to maximize the efficiency of the model, in terms of time, which is why HFT types are developed that can be used in the real world.
- The model must be very accurate, giving the same results as other modern models.
- The model that uses our proposed layer can be interpreted by allowing us to look at what time the learning model arrives. This will help identify future areas of interest where model and horizontal construction can be improved.

III. Methodology

In this section, we evaluate our proposed architecture on the mid-price movement prediction problem based on a large scale high-frequency LOB dataset. Before elaborating on experimental setups and numerical results, we start by describing the dataset and the prediction task.

A. Dataset

In the stock market, brokers buy and sell stocks through an order-driven system that includes all of the remaining limit orders in the limit book. Limited order is a type of order to buy or sell a certain amount of security at a fixed price or better. The trader must specify the type (buy/sell), price, and volume (the number of shares he wants to sell) in a limited way. Buying and selling limited orders includes two sides of the Limit Order Book (LOB): bidding and inquiry.

In t, the best bid price and the most asked price are defined as the highest bid and the lowest prices in LOB, respectively. When the new order limit arrives, the LOB consolidates and arranges orders on both sides according to the prices offered to place the best bid and competitive price at the first level. As there are many price levels in the order book, we look at the top 10 prices from both sides of the LOB. For more information on prescribed books, see [18].

LOB reflects current purchases and demand for shares at different price levels. Therefore, depending on the availability of LOB data, many analytical and predictive problems can be created, such as order flow distribution, excellent bid distribution, question price, or volatile crisis analysis. for the price.

The average price for a given amount of time is defined as the difference between the best bid price and the most competitive price. This price makes sense because no trade is possible at this exact price. Since this quantity is between the best bid area and the most questionable price, its movement reflects LOB strength and market strength. Therefore, the ability to predict future inflation is important. We evaluate our proposed construction to predict the future flow of intermediate prices when a previous bid is granted and request appropriate prices.

For this project, we used a publicly available benchmark database in [19], known as the FI-2010 database. Data were collected from 5 different Finnish stocks on the NASDAQ Nordic from various industry sectors. The collection period runs from 1 June to 14 June 2010, making ten-day order details of approximately 4.5 million events. For each event, the values and volumes of the top 10 orders from each side of the LOB are issued, producing a 40-dimensional vector representation. At the vector of each item, the database includes labels of average values (vertical, up, down) in 5 different horizons. We use this data in our analysis.

B. Bilinear Layer

We begin by introducing some definitions and definitions. In all files the report, we define scalar values either in small or high cases character (a, b, A, B, ...), vectors with lower-case letters (x, y, ...), metrics for bright-faced characters (X, Y,). Matrix X in $R^{D\ timesT}$ is a second-order dual tensor methods with D and T magnitude of the first and second mode mode respectively.

We denote:

X_i $\in$R$^{D\times T}$, where i = 1, . . . , N (Eq. 1)

a set of N samples, each containing the sequence of the previous T visibility corresponding to its T columns. The past tense 'T' values are defined as history when time has a future value The 'H' or 'D' we would like to predict is known as the forecast the horizon.

For example, given that stock prices are always sampled for each second and X_i $\in$R$^{10\times 100}$ contains prices for different LOB levels of the final T = 100 seconds, predictability H = 10 corresponds to prediction a future value, such as Medium price, in the next 10 seconds.

Let's point out X = $[x_1,. . . , x_{T_l}$] $\in$ R $^{D\times T}$ input to Bilinear Layer.

Layer converts input in size D $\times$ T to matrix size D '$\times$ T' using the following map:

$Y = \Phi(W_1XW_2 + B)$ (Eq. 2)

where $W_1 \in R^{D'\times D}$, $W_1 \in R^{T\times T'}$. B $\in$ R$^{D'\times T'}$' are the parameters to estimate.Φ it is flexible element-wise nonlinear function, such as ReLU or sigmoid. One of the obvious benefits of the map in Eq. 2 that the number of parameters is measured in terms of the magnitude of each input mode rather than the number of input neurons.

The most important feature of the map in Eq. 2, if so applied to time series data, that BL models rely on (single with each input mode, representation), each different semantic definitions. To better understand this, describe each one column and X row as X_{c_t} $\in$R^D, t = 1, . . . , T and X_{r_d} $\in$R^T,d = 1,. . . , D, respectively. Given the input series X, t^{th} column represents the exception features or features of the basic process recognized at the time For example t, while the d^{th} line contains temporary variations of the d^{th} feature between previous T steps.

$$\mathbf{W}_1\mathbf{X} = \left[\mathbf{W}_1\mathbf{x}_{c_1}, \ldots, \mathbf{W}_1\mathbf{x}_{c_T}\right] \text{ - (Eq. 3.)}$$

$$\mathbf{X}\mathbf{W}_2 = \begin{bmatrix} (\mathbf{x}_{r_1})^T\mathbf{W}_2 \\ \vdots \\ (\mathbf{x}_{r_D})^T\mathbf{W}_2, \end{bmatrix} \text{ - (Eq. 4.)}$$

Eq. 3 shows that the interaction between the various factors as well features at some point t = 1,. . . , T is taken for W_1 while I was in Eq. 4, W_2 modes a continuation of d^{th} feature or feature. For example, given that X contains stock D values, LOB levels vary between history T, BL determines how different prices come together at a certain time is W_1 and how prices for a specific index continue over time in W_2. It has been shown to take advantage of location the existing structure in LOB produces a better distribution in conjunction with best bid of the future and ask for prices.

C. Temporal Attention Augmented Bilinear Layer

Although BL learns different dependencies with each mode, it is not clear how the representations interact simultaneously with other time conditions or what are the critical time periods in the forecast on T '. By entering the position details in the attention-calculating calculator scheme, we saw that the studied model used only for a certain period of time in the past to predict the future value in a given horizontal study sequence. To learn the value of each time in the proposed BL, we suggest that the overridden Bilinear Layer (TABL) map input X $\in$R$^{D\times T}$ to the output Y $\in$R$^{D'\times T'}$ as follows:

$$\bar{\mathbf{X}} = \mathbf{W}_1\mathbf{X} \text{ - (Eq. 5.)}$$

$$\mathbf{E} = \bar{\mathbf{X}}\mathbf{W} \text{ - (Eq. 6.)}$$

where α_{ij} and e_{ij} mean something in it (i, j) of A and E, respectively , $\odot$ it means duplication of wisdom operator, and Φ is a non-linear map defined as Eq. 2 $W_1 \in R^{D\times T}$, W $\in$R$^{T\times T'}$, $W_2 \in R^{D\times T}$, B $\in$R$^{D'\times T'}$ and λ are a proposed bilinear layer of Temporary Extension. It's like Mentioned above for the Bilinear framework, an additional temporary

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T} \exp(e_{ik})} \quad \text{- (Eq. 7.)}$$

$$\tilde{\mathbf{X}} = \lambda(\bar{\mathbf{X}} \odot \mathbf{A}) + (1-\lambda)\bar{\mathbf{X}} \quad \text{- (Eq. 8.)}$$

Bilinear Layer models depend on different W1 and W2 variants for the inclusion of intermediate attention step read W and λ.

To proceed with the Temporal Augmentation bilinear layer we have 5 steps, which are described in detail as follows:

• In Eq. 5, w_1 is used to change the representation each time X_{ct}, where t = 1,. . . , T of X (per column) in the new feature space $R^{D'}$. This model relies on the original X mode while keeping the temporary setting inactive.

• The second step aims to learn how important temporary conditions are to each other. This is achieved by reading a structured matrix W with dense elements centered on 1 / T. Let's explain $\bar{X}_t$ $\in$R$^{D'}$ and e_t $\in$R$^{D'}$ the t column X and E respectively. From Eq. 6, we see that e_t is a weighted combination of T-temporal positions in the space of $R^{D'}$, i.e., T-columns of X, which have periods of time always equal to 1 / T since the diagonals of W are set to 1 / T. Therefore, the element e_{ij} in E adds the corresponding value of the $\bar{x_{ij}}$ item to the other - $\bar{x_{ik}}$, where k $\neq j$.

• Normalize the values of E using the softmax function in Eq. 7, the proposed layer pushes many objects to close to zero while keeping prices some of them positive. This process produces the attention mask A.

• Attention mask A found in step three is used to eliminate the effect of non-essentials on $R^{D'}$. Instead of using the hard-earned approach, the readable scale λ in Eq. 8 allows the model to learn a soft attention span. In the first stage of the learning process, the learning features extracted from the previous layer can be noisy and non-discriminatory, so hard attention can mislead the model to insignificant information. In contrast, soft attention may allow the model to learn discriminatory features at the beginning of the phase. Here we must know that it is compulsory to sleep in it width [0, 1], i.e. 0 λ 1.

• Similar to Bilinear Layer, the final step of the proposed layout estimates the w_2 interim map, excluding high-level representation after a change of bias and linearity.

In general, the introduction of a focused approach in the second, third and fourth steps of the proposed layer promotes competition between neurons representing different temporal steps of the same factor, i.e., competition between objects in the same line of $\bar{x}$. Competitions, however, are independent of each element in the $R^{D'}$, i.e. items in the same $\bar{x}$ column do not compete to be represented. The proposed layer construction is trained in conjunction with other layers in the network using the Back-Propagation algorithm.

$$\mathbf{Y} = \phi(\tilde{\mathbf{X}}\mathbf{W}_2 + \mathbf{B}) \quad \text{- (Eq. 9.)}$$

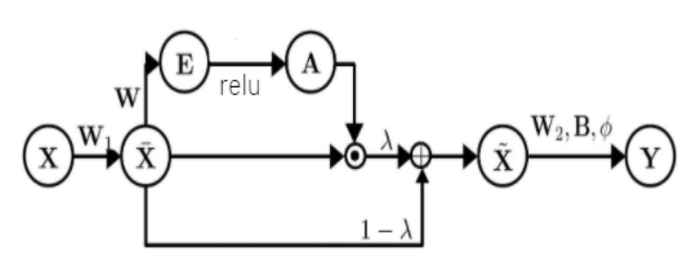

Fig. 1. Temporal Attention Augmentation Bilinear Layer Topology

D. Model Summary

```
Model: "model_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_3 (InputLayer)         (None, 40, 10)            0
_________________________________________________________________
tabl_7 (TABL)                (None, 60, 10)            3201
_________________________________________________________________
activation_7 (Activation)    (None, 60, 10)            0
_________________________________________________________________
dropout_5 (Dropout)          (None, 60, 10)            0
_________________________________________________________________
tabl_8 (TABL)                (None, 120, 5)            7951
_________________________________________________________________
activation_8 (Activation)    (None, 120, 5)            0
_________________________________________________________________
dropout_6 (Dropout)          (None, 120, 5)            0
_________________________________________________________________
tabl_9 (TABL)                (None, 3)                 394
_________________________________________________________________
activation_9 (Activation)    (None, 3)                 0
=================================================================
Total params: 11,546
Trainable params: 11,546
Non-trainable params: 0
_________________________________________________________________
```

Fig. 2. Model Architecture of Temporal Augmentation Bilinear Layer

The above model describes the neural network layers formed by the bilinear temporary attention layer consisting of an X input scale X × 40 × 10, followed by 2 pairs of BL layer and Dropout layer that randomly assigns 0 input units to each training frequency, which helps prevent overlap of 60 × 10 and 120 × 5 sizes respectively, using a temporary extension of the 3 × 1 size extension. Finally, the output layer is used to produce the desired map Y. The reason why the extra layer of temporary attention was used in the third layer was to promote competition between objects from time to time. one $T_o \in T$
or similar guess limit $D_o \in D$
which can improve the performance of the model.

We used the Cross-Entropy section to calculate losses in training and test modeling with FI-2010 data. This has helped us train the bilinear layer temporary care model to deliver opportunities over 3 stages in each data event.

The model's efficiency was done with the help of ADAM, which helped us integrate the excellent AdaGrad architecture with RMSProp algorithms to provide an optimization algorithm that could handle small gradients in a sound database such as the FI-2010 dataset.

E. Novelty

When we studied the model presented in the base paper, we noticed that the authors had proposed using the main temporal attention augmentation mechanism only in the final layer, while the rest of the layers were normal bilinear layers only.

We decided to try to improve upon this model by incorporating the temporal attention into all the hidden layers. We argue that any minor decrease in the time efficiency of the model will be offset by the its improved accuracy. As we present in the results and analysis section, we have got some positive results in that regard.

IV. Results and Analysis

In this project, we used a built-in learning model to identify short-term indicators from the data and used it to distinguish whether the stock price would increase, decrease or remain constant over the forecast. In this study, we took the predictive horizon, k = [10, 20]. Here, the value of k indicates the number of past events in which the model predicts the state (ups, downs, or stops) of the average stock price.

For this project, we used a standard 7: 3 data division. In our case, it means that the first 7 days data was used as training data while the last 3 days data was used as test data. Specifically, the training set consists of 2.54 lakh samples, and the test set contains 1.39 lakh samples.

While the base paper we used provided a prediction of k = 50 and 100, due to the hardware and software limitations we have, we were unable to test our model at those values (overloaded RAM). However, we are confident that our model will surpass the results associated with the basic paper.

Due to the nature of its real-world, the database does not match the bulk of the standing class samples. Therefore, we adjusted the hyper-parameters according to the average F1 rating per class, which is a trade between accuracy and memory, measured in a training set.

For analysis, we compare the results of our improved model compared with the base paper we used with the most recent paper entitled, DeepLOB [21].

Predictability estimates k = 10; below is the classification report we received. Due to data inequalities, we use tools to amplify data, which is why we use a limited scale for analytical purposes.

	precision	recall	f1-score	support
0	0.70	0.72	0.71	38464
1	0.83	0.80	0.82	66002
2	0.68	0.71	0.70	35112
accuracy			0.76	139578
macro avg	0.74	0.74	0.74	139578
weighted avg	0.76	0.76	0.76	139578

Fig. 3. Results of model for k = 10

As is clear, we are getting an F1 score of 76% for our model. For comparison purposes, the base paper authors got an F1 score of 77.63% while the DeepLOB model gives a score of 83.40% for similar parameters.

While the score difference between our model and the base paper is very less, the score difference between our model and the DeepLOB paper is due to the fact that the model described in that paper utilized more that 10 layers of CNN and LSTM layers, each of which contain 64 nodes. Also, since our model is based on temporal cues, it has very little information to base its predictions on for small prediction horizons such as k = 10 or lower.

For the prediction horizon of k = 20, below is the classification report which we obtained.

	precision	recall	f1-score	support
0	0.70	0.66	0.68	38454
1	0.81	0.80	0.81	66002
2	0.66	0.71	0.68	35112
accuracy			0.74	139568
macro avg	0.72	0.72	0.72	139568
weighted avg	0.74	0.74	0.74	139568

Fig. 4. Results of model for k = 20

As is clear, we are getting an F1 score of 74% for our model. For comparison purposes, the base paper authors got an F1 score of 66.93% while the DeepLOB model gives a score of 72.82% for similar parameters.

Therefore, our model performs far superior to the base paper implemented and slightly better than the latest model proposed in the field for the prediction horizon of k = 20.

V. Conclusions and Future Work

In this project, we have built a custom-built deep learning model which classifies if the price of a stock in a limit order book increases, decreases or remains stationary during high frequency trading.

Our model is quite efficient and accurate, with only 2 hidden layers, especially with respect to the other state-of-the-art models in the market such as DeepLOB, which utilize a far greater number of layers with classical Deep Learning models such as CNN and LSTM.

With regard to future work, we would like to work on this model to increase the effectiveness for lower prediction horizons by combining works such as the DeepLOB model and our model to improve overall performance and achieve effective compromise between time-efficiency and accurate results.

Acknowledgments

We would like to express our sincere gratitude t our supervisors, Dr. Sowmya Kamath and Dr. Anand Kumar, for their enthusiasm, patience, insightful comments, helpful information, practical advice and unceasing ideas that have helped us tremendously at all times towards the creation of this project.

References

[1] A. Graves, A.R. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” in Acoustics, speech and signal processing (icassp), 2013 ieee international conference on, pp. 6645–6649, IEEE, 2013.

[2] S. B. Taieb and A. F. Atiya, “A bias and variance analysis for multi step ahead time series forecasting,” IEEE transactions on neural networks and learning systems, vol. 27, no. 1, pp. 62–76, 2016.

[3] G. E. Box, G. M. Jenkins, and J. F. MacGregor, “Some recent advances in forecasting and control,” Applied Statistics, pp. 158–179, 1974.

[4] G. C. Tiao and G. E. Box, “Modeling multiple time series with applications,” journal of the American Statistical Association, vol. 76, no. 376, pp. 802–816, 1981.

[5] J. J. Hopfield, “Neural networks and physical systems with emergent collective computational abilities,” in Spin Glass Theory and Beyond: An Introduction to the Replica Method and Its Applications, pp. 411415, World Scientific, 1987. [6] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al., “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” IEEE Signal Processing Magazine, vol. 29, no. 6, pp. 82–97, 2012.

[7] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, “Beyond short snippets: Deep networks for video classification,” in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4694–4702, 2015.

[8] M. Riemer, A. Vempaty, F. Calmon, F. Heath, R. Hull, and E. Khabiri, “Correcting forecasts with multifactor neural attention,” in International Conference on Machine Learning, pp. 3010–3019, 2016.

[9] A. Tsantekidis, N. Passalis, A. Tefas, J. Kanniainen, M. Gabbouj, and A. Iosifidis, “Forecasting stock prices from the limit order book using convolutional neural networks,” in Business Informatics (CBI), 2017 IEEE 19th Conference on, vol. 1, pp. 7–12, IEEE, 2017.

[10] A. Tsantekidis, N. Passalis, A. Tefas, J. Kanniainen, M. Gabbouj, and A. Iosifidis, “Using deep learning to detect price change indications in financial markets,” in Signal Processing Conference (EUSIPCO), 2017 25th European, pp. 2511–2515, IEEE, 2017.

[11] J. Heaton, N. Polson, and J. Witte, “Deep portfolio theory,” arXiv preprint arXiv:1605.07230, 2016.

[12] A. Sharang and C. Rao, “Using machine learning for medium frequency derivative portfolio trading,” arXiv preprint arXiv:1512.06228, 2015.

[13] Y. Deng, F. Bao, Y. Kong, Z. Ren, and Q. Dai, “Deep direct reinforcement learning for financial signal representation and trading,” IEEE transactions on neural networks and learning systems, vol. 28, no. 3, pp. 653–664, 2017.

[14] Y. Deng, Y. Kong, F. Bao, and Q. Dai, “Sparse coding-inspired optimal trading system for hft industry,” IEEE Transactions on Industrial Informatics, vol. 11, no. 2, pp. 467–475, 2015.

[15] J. Sirignano, “Deep Learning for Limit Order Books,” ArXiv e-prints, Jan. 2016.

[16] W. Bao, J. Yue, and Y. Rao, “A deep learning framework for financial time series using stacked autoencoders and long-short term memory,” PloS one, vol. 12, no. 7, p. e0180944, 2017.

[17] R. Cont, “Statistical modeling of high-frequency financial data,” IEEE Signal Processing Magazine, vol. 28, no. 5, pp. 16–25, 2011.

[18] A. Ntakaris, M. Magris, J. Kanniainen, M. Gabbouj, and A. Iosifidis, “Benchmark dataset for mid-price prediction of limit order book data,” arXiv preprint arXiv:1705.03233, 2017.

[19] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in Proceedings of the IEEE international conference on computer vision, pp. 1026–1034, 2015.

[20] D. T. Tran, A. Iosifidis, J. Kanniainen and M. Gabbouj, ”Temporal Attention-Augmented Bilinear Network for Financial Time-Series Data Analysis,” in IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 5, pp. 1407-1418, May 2019, doi: 10.1109/TNNLS.2018.2869225.

[21] Z. Zhang, S. Zohren and S. Roberts, ”DeepLOB: Deep Convolutional Neural Networks for Limit Order Books,” in IEEE Transactions on Signal Processing, vol. 67, no. 11, pp. 3001-3012, 1 June1, 2019, doi: 10.1109/TSP.2019.2907260.