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*—The noisy and volatile nature of the stock market makes financial time-series analysis a highly challenging topic in the field of deep learning. This has often been attributed to the noisy and volatile nature of the stock market. Especially, in the field of High-Frequency Trading (HFT), prediction is a very challenging task because the automated inference system requires both precision and speed. In this project, we have used an unconventional construction method for predicting time series data, with positive results. These technologies are trained and tested with the benchmark LOB FI-2010 dataset, and the corresponding results are compared and analyzed using a variety of methods.**

***Index Terms*—Temporal attention , shallow neural networks, financial time-series, prediction horizon, attention mask, limit order book.**

I. Introduction

Time-series analysis and prediction has been a widely studied problem in the preceding decades. Time-series analysis has been applied to problems in fields as varied as natural language processing, finance and economics, meteorology, human behaviour analysis and a range of of other fields. Moreover,the complex dynamics of financial markets results in highly non-stationary and noisy observed data. Hence, this represents a very limited perspective of the actual price generating process.

Over the decades, several types of mathematical features have been proposed to extract meaningful and useful data from the stock market time-series data and systems. Examples of such features include the Auto-Regressive Integrated Moving Average (ARIMA) [3], [4]. However, these type of features are often made with several base assumptions, 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 temporal data such as financial data into time series data. However, the type neural network called Recurrent Neural Networks (RNN) is specially designed to extract temporal data from the raw sequential 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 layers by introducing attention mechanism to the temporal mode. The LOB benchmark dataset, the FI-2010 database, was used in this project.

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.

In financial time-series analysis, portfolio trading models were derived using Deep Belief Networks and Auto Encoders

in [7], [8]. A 3 hidden layer MLP (Multi-Layer Perceptron) modelling the joint distribution of bid and ask prices was also used to study the spatial relations between LOB levels in [9].

Many deep neural networks for financial time-series analysis were proposed within a complex forecasting pipeline to account for the noisy and volatile nature of the market. In context of high-frequency LOB data (which is what we use as a dataset), the authors proposed to normalize the LOB states by the preceding days' statistics. These normalized LOB states are then fed into a CNN [10] or an LSTM network [11]. Finally, the authors in [21] proposed a DeepLOB model which made heavy use of multiple CNN layers, before wrapping them in LSTM layers for reinforced learning.

B. Motivation

Most of the deep learning models in the market for High Frequency Trading(HFT) at the moment consist of non-specific classical models such as SVM, CNNs or recurrent structures such as LSTMs. Other custom-made DL models are very deep in nature or are not efficient, both of which are huge disadvantages in the highly competitive field of HFT.

C. Problem Statement

To create a custom-built deep learning model with two hidden layers to classify whether the stock price will increase, decrease or remain stationary over some prediction horizon. This is done by leveraging the idea of bilinear projection and incorporating an attention mechanism in the temporal mode, for maximum efficiency.

D. Objectives

- To create a model with an extremely shallow layered architecture so as to maximise the efficiency of the model, with respect to time, hence building practicable HFT models which can be used in the real world.
- The model must be very accurate, giving results comparable to other state-of-the art models.

III. Methodology

In this section, we examine our proposed structure in the problem of predicting medium price movements based on large LOB high-level databases. Before specifying in the test settings and numerical results, we first define the data and predictive function.

A. Temporal Attention Augmented Bilinear Layer

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

$$\bar{\mathbf{X}} = \mathbf{W}_1\mathbf{X}$$

Fig. 1. - (Eq. 5.)

$$\mathbf{E} = \bar{\mathbf{X}}\mathbf{W}$$

Fig. 2. - (Eq. 6.)

input X $\in \mathrm{R}^{D\times T}$ to the output Y $\in \mathrm{R}^{D'\times T'}$ as follows:

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 $\in \mathrm{R}^{T\times T'}$, $W_2 \in R^{D\times T}$, B $\in \mathrm{R}^{D'\times T'}$ and λ are a proposed bilinear layer of Temporary Extension. An additional temporary Bilinear Layer models depend on different W1 and W2 variants for the inclusion of intermediate attention step 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 in the new feature space $R^{D'}$. This model relies on the 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 $\bar{X}_t$ $\in \mathrm{R}^{D'}$ and e_t $\in \mathrm{R}^{D'}$ 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.

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T} \exp(e_{ik})}$$

Fig. 3. - (Eq. 7.)

$$\tilde{\mathbf{X}} = \lambda(\bar{\mathbf{X}} \odot \mathbf{A}) + (1 - \lambda)\bar{\mathbf{X}}$$

Fig. 4. - (Eq. 8.)

• 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.

B. Model Summary

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 dataset such as the FI-2010 dataset.

C. 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

$$\mathbf{Y} = \phi\left(\tilde{\mathbf{X}}\mathbf{W}_2 + \mathbf{B}\right)$$

Fig. 5. - (Eq. 9.)

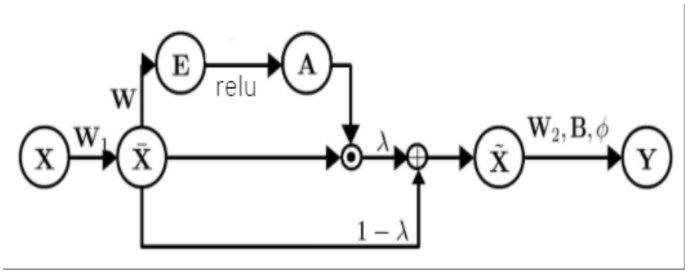

Fig. 6. Temporal Attention Augmentation Bilinear Layer Topology

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. 7. Model Architecture of Temporal Augmentation Bilinear Layer

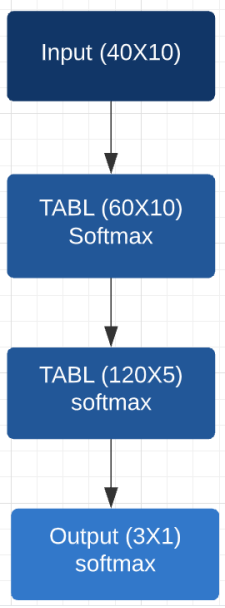

Fig. 8. Model Topology

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.

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

	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. 9. Results of model for k = 10

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. 10. 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.