



HAL
open science

Fine-Grained Double-View Link Prediction Within the Dynamic Interaction Network

Jianye Pang, Wei Ke

► **To cite this version:**

Jianye Pang, Wei Ke. Fine-Grained Double-View Link Prediction Within the Dynamic Interaction Network. 18th IFIP International Conference on Artificial Intelligence Applications and Innovations (AIAI), Jun 2022, Hersonissos, Greece. pp.257-269, 10.1007/978-3-031-08333-4_21 . hal-04317197

HAL Id: hal-04317197

<https://inria.hal.science/hal-04317197v1>

Submitted on 1 Dec 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



This document is the original author manuscript of a paper submitted to an IFIP conference proceedings or other IFIP publication by Springer Nature. As such, there may be some differences in the official published version of the paper. Such differences, if any, are usually due to reformatting during preparation for publication or minor corrections made by the author(s) during final proofreading of the publication manuscript.

Fine-Grained Double-View Link Prediction within the Dynamic Interaction Network

Jianye Pang and Wei Ke^(✉)

Xi'an Jiaotong University, Xi'an, Shaanxi, P.R.China
jiane.pang97@gmail.com, wei.ke@mail.xjtu.edu.cn

Abstract. Trend prediction in financial trading markets is crucial in decision-making support, normally realized via mining trading patterns or analyzing technique indicators. However, cryptocurrency decentralized exchange (DEX) markets are growing rapidly recently and draw much attention. Different from the centralized market, DEX allows users to observe the whole market data at the order-level, where identity information associated with orders is also available. Moreover, trading pairs in DEX are often manipulated by bots, and user ordering behavior may be affected by these algorithm-controlled bots. These provide us with a novel human-bot game scenario to investigate the predictability of human ordering behavior. In this paper, we model the trading market of the cryptocurrency DEX at each timestamp as a user-order bipartite graph and propose a double-view dynamic network for fine-grained user ordering prediction in a human-bot mixed DEX market. Given the ordering information of the humans and bots, we learn current latent representations of humans and bots using graph neural networks and recurrent neural networks respectively for user ordering prediction. We conduct experiments on real cryptocurrency data from Binance Dex. The experimental results of link prediction demonstrate that our proposed method outperforms the state-of-the-art models.

Keywords: Cryptocurrency decentralized exchange · User-order link prediction · Bot market manipulation · Dynamic network.

1 Introduction

Market trend prediction is to predict the price fluctuations in the financial market for financial analysis. It is of great significance for reducing investment risk and making financial decisions. Despite the successful machine learning techniques [26, 10, 9, 2] in predicting traditional market price trends, these methods tried to represent the market performance only from a macro-level, either with indicators describing aggregated behavior from users [10, 9] or methods modeling markets across time periods [26, 2]. However, most existing studies are designed for centralized exchanges (e.g., stock), where public-available orders cannot be associated with a specific user. This prevents studies on market prediction from the perspective of fine-grained user ordering behavior.

In this study, we are the first to investigate the market trend prediction in decentralized exchanges, i.e., the prediction of which users will place orders at what price in the very near future of a trading pair. Benefiting from the openness, transparency, and credibility of blockchain technology [20, 21], decentralized exchanges [12] (DEX) can overcome the problem of opaque trading data [20, 21]. Moreover, the fine-grained ordering information with user identity can be directly obtained from DEX OrderBooks [20].

However, there are some unique challenges in DEX market prediction. Firstly, the ordering information with user identity allows us to associate specific orders with their users. It needs to explore the representation of user-order dynamic interactions in the cryptocurrency market. Secondly, in order to improve their rankings and prominence, exchanges support using bots to manipulate markets [15, 1]. In this new type of bot-human game market, bot behavior misleads the user’s ordering intentions and trading decisions, so as to influence the market trend. Therefore, properly modeling how bots influence user behavior is critical to making accurate predictions.

To address the challenges mentioned above, in this paper, we develop a novel double-view dynamic network framework at the fine-grained timestamp for user ordering prediction in a human-bot mixed DEX market. Specifically, given the real data from Binance DEX ¹, at each timestamp, we model the trading market of the current cryptocurrency DEX as a user-order bipartite graph and propose a dynamic network using graph neural network (GNN) and recurrent neural network (RNN) models to represent users, placed orders, and temporal user-order links, where the GNN is designed to model long-term user ordering characteristics in the past time, and the RNN is used to model the instantaneous user attribute variation and ordering information at the current timestamp. Besides, we design a joint training strategy for the dynamic network to acquire final user ordering predictions. The experimental results of link prediction demonstrate our model surpasses other state-of-the-art methods by a large margin. We also conduct extensive ablation results to verify the robustness of our model in market trend prediction.

The paper is organized as follows: Section 2 summarizes related work on the traditional stock market and dynamic network. Section 3 details the formulation of a user-order bipartite graph, the process of double-view dynamic network framework at fine-grained timestamps and the joint training strategy. Section 4 presents experimental results and analysis of link prediction. Lastly, the final conclusions are discussed in Section 5.

2 Related Work

Traditional Stock Market Analysis. Traditional stock market researches require a thorough understanding of the market and a clear investment strategy [18, 6]. Based on the assumption [4] that stock investors seek profit maximization strategies by trying to discover the potential trading patterns to predict

¹ <https://www.binance.org/en>

the future price trend, it mainly focuses on the trading patterns and the trend prediction of price fluctuations. The SFM [26] model proposed by Zhang et al. aims to clearly discover and distinguish different frequencies of potential trading patterns to predict the stock price using the unit gate of RNN. Li et al. [10] designs a framework of technical trading index optimization, which optimized the original technical index leveraging stock attributes. The RNN-MRF [9] model based on multi-task learning achieves the prediction of price rise or decline by introducing technical indicators.

Continuous Dynamic Network Representation. At present, there are two major paradigms used for network modeling in continuous dynamic network research: dynamic graph neural network (DGNN) and time series [22, 7]. DGNN-based network representations mainly introduce timing rules to node aggregation so as to obtain dynamic graph embedding. CTDNE [14] and TDGNN [17] use the time-dependent aggregation method to aggregate the features of adjacent nodes and the time information of the edges to obtain the representation of the target node. Although there are other methods of dynamic graph neural networks, such as EvolveGCN [16] and WD-GCN [13], these methods believe that network changes are discrete time points in the form of the snapshot, which is not suitable for networks with continuous timestamps. Time series-based network representations include RNN and time point process (TPP). In the RNN-based approach, node embeddings are maintained by an RNN-based architecture. Once an event occurs or the network changes, the embedding of interactive nodes will be updated, which enables the embedding to be continuously kept up to date, e.g. the JODIE model [8]. The TPP-based approach, such as the Dyrep [25] and Know-Evolve [24], both use a TPP parameterized by an RNN cyclic architecture.

3 Method

3.1 Preliminaries

Dynamic network Construction To formulate the user ordering prediction as a link prediction task, we first construct the dynamic network $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$. To determine the nodes and links in the dynamic network, the same account address is recorded as the same user, which is the user node; We sort the order by price and divide the price at equal intervals after excluding the abnormal extreme price. Prices in the same interval are considered to be in the same order, which is regarded as the order node. The user node and order node make up all the nodes \mathcal{V} in the network. Therefore, we construct the dynamic network as a user-order bipartite graph, where each link in \mathcal{E} is a user ordering action at a timestamp.

Problem Definition Given a dynamic network $\mathcal{G}_t = \{\mathcal{V}_t, \mathcal{E}_t\}$ at the current timestamp t which involves a set of observed user-order links $\mathcal{E}_t = \{(u, v) \mid u, v \in \mathcal{V}_t\}$ which happened before the next timestamp $t + 1$, our goal is to predict those unobserved user-order links in \mathcal{G} at timestamp $t + 1$. We treat this task as binary classification, that is, given a pair of nodes (u, v) , we predict if there is a user-order link between them at the next timestamp $t + 1$.

3.2 Fine-Grained Double-View Framework

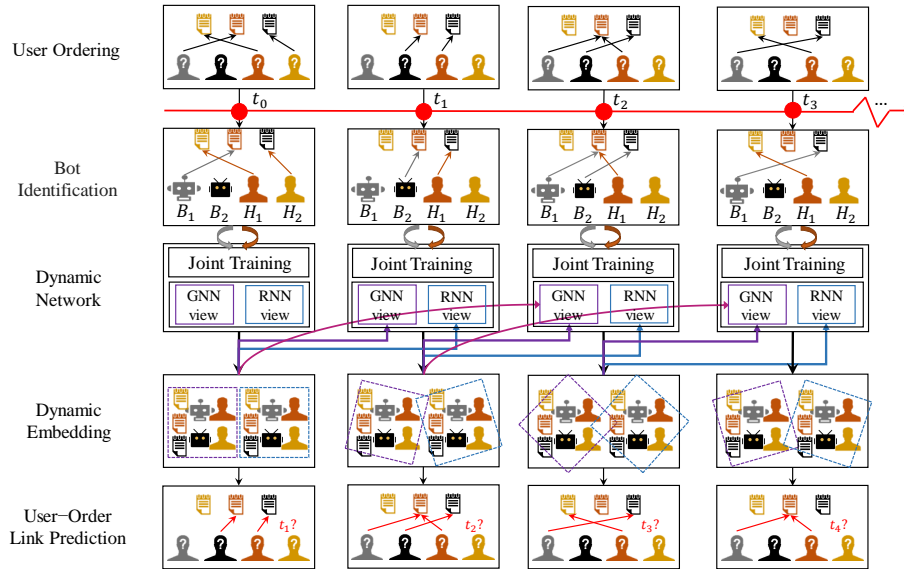


Fig. 1. The overview of our proposed fine-grained double-view framework. B denotes the bot and H denotes the human.

To predict user ordering, we propose a novel fine-grained double-view framework to predict user-order links in a dynamic network. Figure 1 is an illustration of our proposed framework. At each timestamp, firstly, we follow three strong guidelines detailed in the financial research [3] to filter and identify bot addresses as a rough pre-processing. Secondly, we develop a hybrid GNN-RNN-based dynamic network representation method for the user and order embedding at the current timestamp. The GNN is designed to capture network topology changes by considering long-term user ordering information in the global, and the RNN is used to model the instantaneous user drifting and ordering information in the local by the user-order interactions at the current timestamp. The whole framework is trained jointly. Finally, given the updated user and order embedding results in the current network, we predict potential user-order links at the next timestamp with the fully connected network.

3.3 Dynamic Network Representation

Multi-view Dynamic Embedding We adopt the dynamic embedding of GNN and RNN, which is multi-view and goes into detail as below.

GNN View Dynamic Embedding: Our GNN branch consists of continuous-time candidate neighbors selection, temporal context windows and exponential time-dependent aggregation to obtain GNN’s dynamic embedding.

When we update the current node u using neighbor nodes that have historical interactions, we are biased against different neighbor nodes. Here we express the bias in the selection of neighbor nodes when the human node is updated. We suppose the current ordering is more affected by the market price at recent timestamps. For the order occurring at a very early time, we think it causes negative interference, thus we only need to decide the neighbor nodes $N(u) = \{w_\tau\}$, τ is a timestamp t^- near the current timestamp t .

Continuous-time candidate neighbors selection We use an exponential function to calculate sampling probability and decide whether neighbor nodes are sampled in this update. Given a target node u at timestamp t , each candidate neighbor $w \in N_t(u)$ has the following probability $p(w)$ of being selected:

$$p(w) = 1 - \frac{e^{t-\tau(w)}}{\sum_{w' \in N_t(u)} e^{t-\tau(w')}} \quad (1)$$

We suppose that the neighbors at the nearer timestamp interacting with the current target node are more likely to be selected as candidate neighbors. Because at the nearer timestamp, the more likely the buying and selling order price will be repeated soon. The longer the time span is, the less likely it is. After we get all probabilities of candidates, we choose n nodes with the highest probabilities as the neighbors.

Temporal context windows Since we need an expected reasonable minimum number of candidates to update the dynamic network, we set a temporal context window for restriction. We suppose the window size can maintain the ordering interest shifts and also obtain sufficient information about the deformation structure of the graph. Therefore, we set the fixed window H to limit the scope of candidates and use the neighbor selection strategy to capture valid neighbors appearing inside the window. Therefore, the first optional time point t_0 is chosen as $t_0 = t - H + 1$.

Exponential Time-dependent Aggregation We adopt an exponential time aggregator in the GNN branch to encourage the GNN model to concentrate on local context information. For simplicity, we describe it in a single-layer forward aggregator and it can be easily generalized to multi-layer networks. Assuming the initial target node representation is \vec{h}_u^0 , the new node representation at time t can be calculated as follows:

$$\vec{h}_u^k = \sigma \left(\sum_{w \in N_t(u) \cup u} \alpha_{uw}^t W \vec{h}_w^{k-1} \right) \quad (2)$$

where k represents the k -th layer of the time aggregator, σ is a nonlinear activation function such as ReLU, W is the learnable shared weight matrix and the neighbor nodes aggregating weights α_{uw}^t at time t can be calculated as follows:

$$\alpha_{uw}^t = 1 - \frac{e^{t-t_{u,w}}}{\sum_{w \in N_t(u) \cup u} e^{t-t_{u,w}}} \quad (3)$$

where $t_{u,w}$ denotes the timestamp nearest to t at which link (u, w) interacts.

RNN View Dynamic Embedding: Inspired by JODIE [8], our RNN branch consists of embedding update, projection and predicting next order embedding to obtain RNN’s dynamic embedding.

Embedding update RNN branch uses a coupled RNN to update both human and bot links. RNN_U updates dynamic user embedding and RNN_O updates dynamic order embedding which are parameter shared among all users and orders respectively.

When user u interacts with order o , RNN_U updates user embedding $u(t)$ using the order embedding $o(t^-)$ near time t . In the same way, RNN_O updates order embedding $o(t)$ using the user embedding $u(t^-)$. Both embedding updates are represented as follows:

$$\begin{aligned} \mathbf{u}(t) &= \sigma(W_1^u \mathbf{u}(t^-) + W_2^u \mathbf{o}(t^-) + W_3^u \mathbf{f} + W_4^u (t - t_u^-)) \\ \mathbf{o}(t) &= \sigma(W_1^o \mathbf{o}(t^-) + W_2^o \mathbf{u}(t^-) + W_3^o \mathbf{f} + W_4^o (t - t_o^-)) \end{aligned} \quad (4)$$

where f is the feature of user-order interaction links and W_i is the learnable matrix. Note that for the user-order link, we set the feature f to extract according to the actual semantics of the user ordering action. The feature consists of the actual price of the order, the quantity of the order, the current market price, order types (buy or sell), other valuable numerical data from OrderBook and one-hot encoding of categorical information at this timestamp, etc.

Embedding projection To predict the future embedding at timestamp t^+ , we project the user embedding after the time t of the last user interaction. The projection embedding is the product of a scaled linear factor W_s and the previous user embedding $u(t)$ as $\tilde{u}(t^+) = W_s u(t)$. The greater value of t^+ is, the greater drifting of the projected embedding vector is over time. As t^+ increases, mapped embedding drifts more severely.

Predicting next order embedding Following JODIE’s setting, we train the RNN branch to minimize the L_2 difference between the predicted order embedding $\tilde{o}(t^+)$ computed by using $\tilde{u}(t^+)$ and the real order embedding $\bar{o}(t^+)$ as $\|\tilde{o}(t^+) - \bar{o}(t^+)\|_2$. Therefore, for a given user, we can retrieve the order most likely to interact with at the current timestamp. Since the query is $O(1)$, we can easily query all users to find their potential user-order links where we achieve the "double-end" link prediction leveraging the "single-end" prediction.

Joint Training Strategy We fuse and rerank the double-view results of GNN and RNN branches at each timestamp to get the final user-order link prediction results at the current timestamp. Specifically, from the GNN view, for the potential link pair (u, v) in the graph, we use the weighted-L1 link aggregation function to obtain the link representation as $|h_v - h_u|$. From the RNN View, for each potential link pair (u, v) in user and order embedding matrices, the link representation is computed as the concatenation $[h_v, h_u]$ of their corresponding embeddings.

For every branch, we train a fully connected network separately and send their user-order link representations respectively to compute the prediction of whether they happen at this timestamp in each view which is regarded as binary classification. Firstly, we rank the Top- k links of each individual multilayer perceptron (MLP) as a preliminary prediction from the GNN view and RNN view according to probabilities from high to low, where k denotes the preset number of candidate links. Secondly, considering both GNN and RNN, the rerank strategy is to combine the total 2k prediction results and sort them by inference probabilities again. If a link appears more than once, the maximum probability of this link is retained.

The GNN branch is trained to minimize the cross-entropy whether the link pair happens at this timestamp as binary classification. The loss of the GNN branch is as follows:

$$\mathcal{L}^G = \sum -(y \log(p) + (1 - y) \log(1 - p)) \quad (5)$$

The RNN branch is trained to minimize the L_2 distance between the predicted order embedding and the ground truth at every interaction, with the same regard to user embeddings and order embeddings. The loss of the RNN branch is as follows:

$$\mathcal{L}^R = \sum (\|\tilde{o}(t) - \bar{o}(t^-)\|_2 + \|u(t) - u(t^-)\|_2 + \|o(t) - o(t^-)\|_2) \quad (6)$$

Hence, we calculate the sum of these two losses in a prediction as follows:

$$\mathcal{L}^S = \mathcal{L}^G + \mathcal{L}^R \quad (7)$$

Due to each timestamp, we need to predict links of the real human and the bot altogether. Here we use a hyperparameter β to balance the modeling power of their ordering behavior. We suppose it is more difficult to learn the prediction of human beings [15, 1], and we need to force the model to better represent real humans. Hence, we re-formulate the total loss as follows to force the distinction between the two objects:

$$\mathcal{L}^S = \mathcal{L}_b^S + \beta \mathcal{L}_h^S \quad (8)$$

4 Experiments

4.1 Datasets

For user-order link prediction, we collect the real data on Binance DEX including the time series of every millisecond OrderBooks and OrderUpdates. Our previous observation of real DEX data shows that when the market price fluctuates fiercely, the number of human traders participating in the market activity increases dramatically. At this time, a large number of real users flood in and place their strategy lists, which is also consistent with our intuition. Therefore, in order to ensure the fairness of the experiment and to prevent the particularity caused by weekends, we select five days in early June 2020, from June 1 to

June 5, from 10 am to 4 pm for 6 hours each day, as our experimental data. For market trading pairs, we choose two representative trading pairs Travalab-Binance Coin (AVA-BNB) and Atomic Wallet Coin-Binance Coin (AWC-BNB) in Binance DEX to analyze since they are long term active and closely related to Binance native token BNB.

To determine nodes and links in the dynamic network for user-order link prediction, the same wallet address is recorded as the same user node. We determine a reasonable minimum interval of prices based on real OrderBooks and past research [23], and each price interval is considered to be the same order node. Starting at 10 a.m. each day, we split the data into half-hourly portions in a non-overlapping sliding window manner. For user-order link prediction, the links that occur in the first 20 minutes of every half-hour are set as the training set, the 20 to 25 minutes as the validation set, and the 25 to 30 minutes as the test set. Therefore, we test models specific on the interactions that occurred in the last one-third of the time for each data portion. Each model is continuously updated within 6 hours of daily data to alleviate potential serious effects caused by remaining missing hours between days.

4.2 Baseline Methods

We compare our method against the following baselines:

- **DynGEM** [5]. This method divides the timeline into discrete time points and learns embedding for the graph snapshots at these time points, which incrementally learns graph autoencoders of varying layer sizes.
- **DySAT** [19]. This method constructs node representations which incorporate the self-attention mechanism into structural neighbors and temporal dynamics to obtain dynamic self-attention network embedding.
- **HTNE** [27]. This method integrates the Hawkes process into deep neural embedding on the neighborhood formation sequence to capture the influence of historical neighbors on the current neighbors.
- **M²DNE** [11]. This method incorporates mutual evolution of micro-dynamics using a temporal attention point process and macro-dynamics using a dynamics equation in a temporal network to model the network structure and alternately affect the process of learning node embedding.
- **JODIE** [8]. This method explicitly models the future trajectory of the nodes in the embedding space for a dynamic recommendation, which defines a projection to predict the dynamic embedding trajectory.

It is worth mentioning that the baselines are all based on network embedding, which includes discrete and continuous network representation of GNN-based and time series-based methods.

4.3 Experimental Settings

Evaluation Metrics. We treat user-order link prediction as a binary classification task. Specifically, with a pair of nodes (u, v) as input, we predict whether

user u will place order v at the next timestamp. Following commonly used settings [16, 8], we leverage two widely-used classification and ranking metrics, namely Mean Reciprocal Rank (MRR) and Recall.

Parameter Settings. For AVA-BNB, there are approximately 20 user nodes, 200 order nodes, and 400 user-order links every half an hour; For AWC-BNB, there are approximately 40 user nodes, 200 order nodes, and 300 links every half an hour. In the parameter setting of the dynamic network model, we set the embedding size of GNN for user and order nodes to 128, and the number of network layer size to 1. The window size is set to 12 and n in neighbor selection is set to 6. We set the dynamic embedding size in RNN to 128. We set β to 1.0 in AVA-BNB and 2.5 in AWC-BNB. For the k in top-k of joint training and N in Recall@N evaluation metric, we record the top 10. The training epoch is generally set to 60.

4.4 Performance on User-Order Link Prediction

In this section, we evaluate all models’ performance on user-order link prediction on Binance DEX with the AVA-BNB market trading pair. We report mean reciprocal rank (MRR) and Recall@10 in Table 1. We draw the following observations.

Table 1. User-order link prediction results of trading pairs AVA-BNB and AWC-BNB. The subscript denotes how the performance of our method compared with the second-best performance.

Model	AVA-BNB		AWC-BNB	
	MRR (%)	Recall@10 (%)	MRR (%)	Recall@10 (%)
DyGEM	51.2	62.5	43.9	57.7
DySAT	55.7	67.3	50.6	65.0
HTNE	35.2	43.8	27.4	38.1
M ² DNE	56.1	67.5	50.3	63.6
JODIE	56.6	69.4	51.2	65.3
Ours	62.4^{+5.8}	74.9^{+5.5}	55.4^{+4.2}	70.7^{+5.4}

In terms of all evaluation metrics, our method has consistently and significantly outperformed all the baselines on both two representative trading pairs AVA-BNB and AWC-BNB. Specifically, our model achieves a great improvement of 5.8% and 5.5% respectively on MRR and Recall for AVA-BNB, and 4.2% and 5.4% respectively on MRR and Recall for AWC-BNB. On the one hand, the GNN view incorporates temporal context information from continuous-time candidate neighbors and time-dependent aggregation, which better captures the global information and topology information of the network. On the other hand, the RNN view learns fine-grained changes and interest drifting in node representation over time and absorbed the characteristics of user ordering at every moment. Therefore, it’s suitable to model the rapid change of graph structure and the situation of high-frequency transactions in the fine-grained time.

4.5 Analysis on Model Robustness

To test the robustness of our model, we analyze how the model performance changes as the length of the time duration that needs to be inferred in the future increases. To be specific, we vary the duration time to 5, 10, 15 and 20 minutes to infer the user-order links. We choose DySAT, JODIE and M²DNE as the baselines since they have relatively competitive performance.

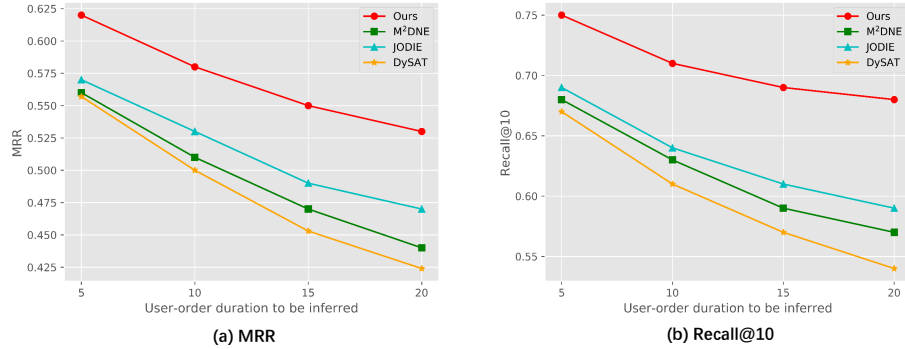


Fig. 2. Performance w.r.t. user-order duration on AVA-BNB.

Our goal is to infer the user-order link at the next timestamp, that is, to keep the model continuously updated over time. This is also in line with the application of trading tools in real scenarios such as the stock market. Due to the dynamic time-varying nature of the cryptocurrency market, with the extension of the duration time, even if the model is always updated, the accumulation of randomness of user behavior at different timestamps will degrade the average inference performance to a certain extent. We report the average performance on different user-order durations to be inferred in Figure 2, where we can see that even as the duration increases, our method still performs the best compared with other baselines, and it has the slowest performance decline, which highlights the robustness of our model in the user-order link prediction task.

4.6 Impact of β Constrained Parameter

To showcase the impact of constrained hyperparameter β used for training, we set β to 1.0, 1.5 and 2.5, where 1.0 means the weights of the bot loss and of the human loss are set by default 1:1, and 2.5 means that we impose a greater penalty on the weight of human ordering prediction. We report the performance with different β values on AVA-BNB and AWC-BNB in Figure 3.

It is obvious that on AVA-BNB, the performance of $\beta=1.0$ is slightly better than 2.5. On AWC-BNB, the performance gets better as the β value increases. One of the most potential reasons is the difference in market trading pairs. Our purpose of β is to balance the learning ability to place orders from humans and bots respectively, and our hypothesis is that it is more difficult to predict orders

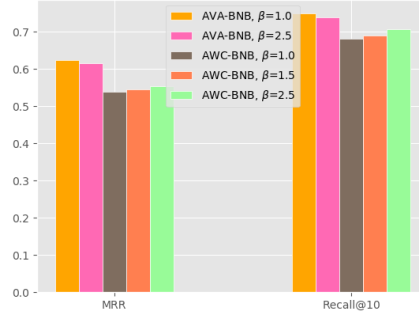


Fig. 3. Performance w.r.t. β constrained parameter.

of real humans, so we add penalty terms on human loss to force the model to focus on human ordering prediction. As a bot-dominated trading market (AVA-BNB), with β increasing, it reduces the modeling power of bot ordering to some extent and slightly lowers the final average performance. As a human-dominated trading market (AWC-BNB), β boosts real human ordering and thus improves the overall performance. We think these are very unique and interesting phenomena in the human-bot mixed cryptocurrency market.

5 Conclusion

This paper formulates user ordering prediction in cryptocurrency DEX as a link prediction task within the dynamic interaction network and proposes a double-view framework at fine-grained timestamps to solve the problem. Compared with baselines, our method shows the proposed framework combines and absorbs the advantages of two graph representations, achieving higher user-order link prediction performance. In addition, our method demonstrates strong robustness through user-order duration experiments. Moreover, compared with multiple market trading pairs, the bot effect on the ordering intentions of real human traders in the human-bot mixed cryptocurrency market is further confirmed.

References

1. Aloosh, A., Li, J.: Direct evidence of bitcoin wash trading. Available at SSRN 3362153 (2019)
2. Chen, C., Zhao, L., Bian, J., Xing, C., Liu, T.Y.: Investment behaviors can tell what inside: Exploring stock intrinsic properties for stock trend prediction. In: ACM SIGKDD (2019)
3. Cong, L.W., Li, X., Tang, K., Yang, Y.: Crypto wash trading. Available at SSRN 3530220 (2020)
4. Dash, R., Dash, P.K.: A hybrid stock trading framework integrating technical analysis with machine learning techniques. *The Journal of Finance and Data Science* **2**(1), 42–57 (2016)
5. Goyal, P., Kamra, N., He, X., Liu, Y.: Dyngem: Deep embedding method for dynamic graphs. arXiv preprint arXiv:1805.11273 (2018)

6. Habib, A., Hasan, M.M.: Business strategy, overvalued equities, and stock price crash risk. *Research in International Business and Finance* **39**, 389–405 (2017)
7. Kazemi, S.M., Goel, R., Jain, K., Kobayev, I., Sethi, A., Forsyth, P., Poupart, P.: Representation learning for dynamic graphs: A survey. *Journal of Machine Learning Research* **21**(70), 1–73 (2020)
8. Kumar, S., Zhang, X., Leskovec, J.: Predicting dynamic embedding trajectory in temporal interaction networks. In: *ACM SIGKDD* (2019)
9. Li, C., Song, D., Tao, D.: Multi-task recurrent neural networks and higher-order markov random fields for stock price movement prediction: Multi-task rnn and higher-order mrfs for stock price classification. In: *ACM SIGKDD* (2019)
10. Li, Z., Yang, D., Zhao, L., Bian, J., Qin, T., Liu, T.Y.: Individualized indicator for all: Stock-wise technical indicator optimization with stock embedding. In: *ACM SIGKDD*. pp. 894–902 (2019)
11. Lu, Y., Wang, X., Shi, C., Yu, P.S., Ye, Y.: Temporal network embedding with micro-and macro-dynamics. In: *CIKM* (2019)
12. Malamud, S., Rostek, M.: Decentralized exchange. *American Economic Review* **107**(11), 3320–62 (2017)
13. Manessi, F., Rozza, A., Manzo, M.: Dynamic graph convolutional networks. *Pattern Recognition* **97**, 107000 (2020)
14. Nguyen, G.H., Lee, J.B., Rossi, R.A., Ahmed, N.K., Koh, E., Kim, S.: Continuous-time dynamic network embeddings. In: *Companion Proceedings of the The Web Conference* (2018)
15. O’Leary, D.E.: Open information enterprise transactions: Business intelligence and wash and spoof transactions in blockchain and social commerce. *Intelligent Systems in Accounting, Finance and Management* **25**(3), 148–158 (2018)
16. Pareja, A., Domeniconi, G., Chen, J., Ma, T., Suzumura, T., Kanezashi, H., Kaler, T., Schardl, T., Leiserson, C.: Evolvegn: Evolving graph convolutional networks for dynamic graphs. In: *AAAI* (2020)
17. Qu, L., Zhu, H., Duan, Q., Shi, Y.: Continuous-time link prediction via temporal dependent graph neural network. In: *Proceedings of The Web Conference* (2020)
18. Sabri, S.R.M., Sarsour, W.M.: Modelling on stock investment valuation for long-term strategy. *Journal of Investment and Management* **8**(3), 60–66 (2019)
19. Sankar, A., Wu, Y., Gou, L., Zhang, W., Yang, H.: Dysat: Deep neural representation learning on dynamic graphs via self-attention networks. In: *WSDM* (2020)
20. Schär, F.: Decentralized finance: On blockchain-and smart contract-based financial markets. Available at SSRN 3571335 (2020)
21. Shapiro, D.C.: Taxation and regulation in decentralized exchanges. *Journal of Taxation of Investments* **36**(1) (2018)
22. Skarding, J., Gabrys, B., Musial, K.: Foundations and modelling of dynamic networks using dynamic graph neural networks: A survey. arXiv preprint arXiv:2005.07496 (2020)
23. Sun, X., Liu, M., Sima, Z.: A novel cryptocurrency price trend forecasting model based on lightgbm. *Finance Research Letters* **32**, 101084 (2020)
24. Trivedi, R., Dai, H., Wang, Y., Song, L.: Know-evolve: Deep temporal reasoning for dynamic knowledge graphs. In: *ICML* (2017)
25. Trivedi, R., Farajtabar, M., Biswal, P., Zha, H.: Dyrep: Learning representations over dynamic graphs. In: *ICLR* (2019)
26. Zhang, L., Aggarwal, C., Qi, G.J.: Stock price prediction via discovering multi-frequency trading patterns. In: *ACM SIGKDD* (2017)
27. Zuo, Y., Liu, G., Lin, H., Guo, J., Hu, X., Wu, J.: Embedding temporal network via neighborhood formation. In: *ACM SIGKDD* (2018)