Shoreline: Data-Driven Threshold Estimation of Online Reserves of Cryptocurrency Trading Platforms (Student Abstract)

Xitong Zhang
Michigan State University
zhangxit@msu.edu

He Zhu
OceanEx Labs, BitOcean Global
hezhu@bitocean.org

Jiayu Zhou
Michigan State University
jiayuz@msu.edu

Abstract

With the proliferation of blockchain projects and applications, cryptocurrency exchanges, which provides exchange services among different types of cryptocurrencies, become pivotal platforms that allow customers to trade digital assets on different blockchains. Because of the anonymity and trustlessness nature of cryptocurrency, one major challenge of crypto-exchanges is asset safety, and all-time amount hacked from crypto-exchanges until 2018 is over $1.5 billion even with carefully maintained secure trading systems. The most critical vulnerability of crypto-exchanges is from the so-called *hot wallet*, which is used to store a certain portion of the total asset online of an exchange and programmatically sign transactions when a withdraw happens. It is important to develop network security mechanisms. However, the fact is that there is no guarantee that the system can defend all attacks. Thus, accurately controlling the available assets in the hot wallets becomes the key to minimize the risk of running an exchange. In this paper, we propose SHORELINE, a deep learning-based threshold estimation framework that estimates the optimal threshold of hot wallets from historical wallet activities and dynamic trading networks.

Introduction

A cryptocurrency transaction is a message propagated in the blockchain network signed by the private key of the sender. When a private key is compromised or stolen, all the funds controlled by the key will be lost. One recent approach aims at solving the problem from the statistical perspective. In (Jain, Felten, and Goldfeder 2018), the authors proposed a threshold control mechanism on the hot wallet to reduce the number of refilling from the cold wallet to the hot wallet, avoiding exposure expectation of cold wallet private keys during transfer, where the cold wallet stores the most of assets offline. Also, the refilling from the offline cold wallet to the online hot wallet is time-consuming. However, it neglects the operational differences among exchanges. The proposed SHORELINE provides a data-driven approach to enable exchange-specific thresholding, by considering historical trading and withdraw/deposit activities in an exchange. There are two major components included as shown in the Figure 1: *a) Dynamic Networks Embedding.* In this component, we embed each cryptocurrency into a low-dimensional vector space. *b) Optimal Threshold Estimation.* To estimate the threshold, we combine multiple data modalities from exchanges, such as the historical trading observations, withdraw and deposit history, currency embedding features. The goal is to predict the threshold of optimal reserves to reduce costly refilling operations and satisfy the online withdraw demand.

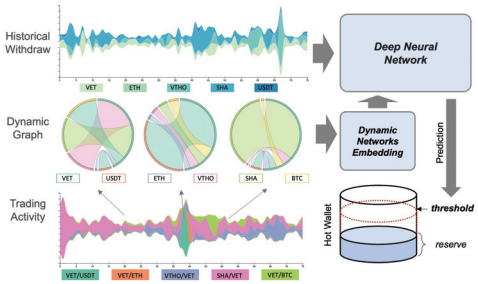

Figure 1: The overview of the proposed SHORELINE framework.

Approach

A trading network G_t at time t is constructed by all trading pairs with their associated trading amount as weights. The temporal network can be considered as the collection of snapshots at different time. The weight is assigned as the trading amount of currency of each trading pair. In order to extract the dynamic pattern from temporal trading networks, we design a contextual embedding network based on node sequences from temporal random walks. For temporal random walks, we create directed temporal edges from historical nodes to their future states between adjacent snapshots. The weight of the edge from node n_t to n_{t+1} is proportional to the weight summation of edges with n_t as one endpoint, where the proportion is a tunable parameter. After temporal integration by introducing temporal edges, we can adopt random walk strategies for static graphs (Mahdavi, Khoshraftar, and An 2018) as temporal random walks. Next, we design a deep neural network based on Bidirectional LSTM mod-

Dataset	IA-Rds	Fb-Frm	Fb-Msg
dyn2v	0.832	0.744	0.663
DynEmb	0.762	0.775	0.764
GRUEmb	0.819	0.820	0.733
LSTM	0.904	0.893	0.787
LSTM_TW	**0.908**	**0.926**	**0.828**

Table 1: AUC scores of link prediction task. The best performance is highlighted.

eling the probability of one sequence from both sides of a node jointly in a classification manner (Devlin et al. 2019). One node might have different embedding features in different contexts, so we take the mean as the final representation.

Then, we design a deep LSTM network to estimate optimal reserves threshold combining trading histories and and embedding features. The output of the estimation framework is the thresholds of the hot wallet. We define the loss function of threshold estimation at a specific time of one sample as: $\mathcal{L} = \sum_i \mathrm{ReLU}(nw_i - \hat{\mu}_i)^2 + \alpha\hat{\mu}_i^2$, where $nw_i = \mathrm{ReLU}(w_i - d_i)$, w_i and d_i are withdraw and deposit amounts of currency i. The first component means that the efficiency cost emerges if the net withdraw amount is higher than the estimated threshold $\hat{\mu}_i$, because we should refill the hot wallet to satisfy the withdraw demand. The second component represents the security concern that all currencies retain online are risky to be stolen. The coefficient α is set to balance the loss from both situations.

Evaluation of Networks Embedding

We start from comparing the dynamic embedding architecture with other baselines by three datasets: IA-Rds (Michalski, Palus, and Kazienko 2011), Fb-Frm (Opsahl 2011), and Fb-Msg (Opsahl and Panzarasa 2009).

Comparison Models: (1) dynn2v (Mahdavi, Khoshraftar, and An 2018). (2) DynGEM (Goyal et al. 2018). (3) GRUEmb (Li et al. 2018). (4) LSTM is the simplified version of our proposed embedding method without temporal random walks. (5) LSTM_TW is our proposed contextual embedding model.

Following (Mahdavi, Khoshraftar, and An 2018), we apply the link prediction task for evaluation. The AUC scores of all testing snapshots are shown in Table 1. The proposed contextual embedding architectures LSTM can achieve better performance than baselines, which supports the efficiency of our proposed contextual embedding. The LSTM_TW outperforms LSTM, achieving the best performance, which proves that sampled sequences from the proposed temporal random walk contain wealthy temporal information. Thus, we further apply it in embedding cryptocurrencies in temporal trading networks.

Evaluation of Shoreline

We use trades of historical six days to predict the threshold of the next day. The evaluation metrics is defined as, $\frac{1}{n}\sum_{i=0}^{n}\sum_{j=0}^{c}\mathrm{ReLU}(nw_{ij} - \hat{\mu}_{ij}) + \alpha\hat{\mu}_{ij}$, where n is the sample size, c is the currency number, different α denotes different levels of concerns for hot wallet security.

Model	$\alpha = 10$	$\alpha = 5$	$\alpha = 3$	$\alpha = 0.1$	$\alpha = 0.01$
OTE-trade	0.361	0.245	0.175	**0.050**	0.020
OTE-hist	0.420	0.333	0.243	0.053	0.017
OTE-emb	0.356	0.278	0.197	0.051	0.019
SHORELINE	**0.240**	**0.177**	**0.143**	**0.050**	**0.015**

Table 2: The table shows the loss metrics based on different α. The best results are highlighted.

Comparison Models: (1) OTE-hist. The input is only historical net withdraws with one layer of LSTM. (2) OTE-trade. The input is raw historical trading records of all trading pairs with one layer of LSTM. (3) OTE-emb. Based on OTE-hist, the other LSTM layer is applied to handle the temporal vectors of digital currencies' embedding. (4) SHORELINE. The final optimal reserves threshold estimation framework combines all the above features and architectures. Different features are fused by a dense layer.

The testing results are listed in Table 2. The SHORELINE can achieve the best testing performance under different security concerns α. The performance of OTE-trade is only slightly better than the historical mean, although the temporal trading networks for embedding are also derived from the trading history. It further support the effectiveness of our proposed embedding method.

Acknowledgments

This material is based upon work supported by the VeResearch research gift from VeChain Foundation, National Science Foundation under Grant IIS-1749940, IIS-1615597, and Office of Naval Research N00014-17-1-2265.

References

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186.

Goyal, P.; Kamra, N.; He, X.; and Liu, Y. 2018. Dyngem: Deep embedding method for dynamic graphs. *arXiv preprint arXiv:1805.11273*.

Jain, S.; Felten, E.; and Goldfeder, S. 2018. Determining an optimal threshold on the online reserves of a bitcoin exchange. *Journal of Cybersecurity* 4(1):tyy003.

Li, T.; Zhang, J.; Philip, S. Y.; Zhang, Y.; and Yan, Y. 2018. Deep dynamic network embedding for link prediction. *IEEE Access* 6:29219–29230.

Mahdavi, S.; Khoshraftar, S.; and An, A. 2018. dynnode2vec: Scalable dynamic network embedding. In *2018 IEEE International Conference on Big Data (Big Data)*, 3762–3765. IEEE.

Michalski, R.; Palus, S.; and Kazienko, P. 2011. Matching organizational structure and social network extracted from email communication. In *Lecture Notes in Business Information Processing*, volume 87, 197–206. Springer Berlin Heidelberg.

Opsahl, T., and Panzarasa, P. 2009. Clustering in weighted networks. *Social networks* 31(2):155–163.

Opsahl, T. 2011. Triadic closure in two-mode networks: Redefining the global and local clustering coefficients. *Social Networks*.