HAL
archives-ouvertes.fr

Profiling Smart Contracts Interactions with Tensor Decomposition and Graph Mining

Jérémy Charlier, Sofiane Lagraa, Radu State, Jerome Francois

▶ **To cite this version:**

HAL Id: hal-01636450

https://hal.inria.fr/hal-01636450

Submitted on 16 Nov 2017

Profiling Smart Contracts Interactions with Tensor Decomposition and Graph Mining

Jérémy Charlier[1], Sofiane Lagraa[2], Radu State[1], and Jérôme François[1,2]

[1] SnT, University of Luxembourg, Luxembourg L-1855, Luxembourg,
[2] Inria Nancy-Grand Est, 54600 Villers-les-Nancy, France
{jeremy.charlier, radu.state}@uni.lu,
{sofiane.lagraa, jerome.francois}@inria.fr

Abstract. Smart contracts, computer protocols designed for autonomous execution on predefined conditions, arise from the evolution of the Bitcoin's crypto-currency. They provide higher transaction security and allow economy of scale through the automated process. Smart contracts provides inherent benefits for financial institutions such as investment banking, retail banking, and insurance. This technology is widely used within Ethereum, an open source block-chain platform, from which the data has been extracted to conduct the experiments.

In this work, we propose an multi-dimensional approach to find and predict smart contracts interactions only based on their crypto-currency exchanges. This approach relies on tensor modeling combined with stochastic processes. It underlines actual exchanges between smart contracts and targets the predictions of future interactions among the community. The tensor analysis is also challenged with the latest graph algorithms to assess its strengths and weaknesses in comparison to a more standard approach.

Keywords: Tensor, Stochastic Process, Graph mining, Smart Contract

1 Introduction

Since the appearance of Bitcoin's crypto-currency in 2009 by Nakamoto, different counterparts have tried to extend Bitcoin's design beyond the currency. The most visible is the Ethereum platform, an open source platform relying on blockchain technology and using Ether as a crypto-currency. They implemented a protocol relying on smart contracts for automatic exchanges without a central gatekeeper. Originally, smart contracts have been first mentioned by Nick Szabo in 1998. He exposed the idea as "*a computerized transaction protocol that executes the terms of a contract [...] to satisfy common contractual conditions, minimize exceptions [and] the need for trusted intermediaries*". Smart contracts appear as a technology for higher transparency without the requirement of trusted intermediaries for currency exchanges. A smart contract is a set of transactions. A transaction has the following fields: *Hash*, *Block_height*, *Sender*, *Receiver*, *Amount* representing the hash of the transaction, the timestamps in terms of block height

numbers, the contract issuer, the contract recipient and the scalar value to be transferred from the sender to the receiver, respectively. Due to the vast number of inter-connected contracts over time, the modeling of the interactions is non trivial and requires an efficient model to predict transactions over time. Predicting and analyzing new contracts has important practical applications in financial instruments where recommendation would be based on likely future contracts interactions.

The main contribution of the paper resides in the modeling of smart contracts interactions using a tensor approach. In addition, a novel technique consisting in the combination of stochastic processes and tensors allows to reproduce existing interactions and to predict future interactions over time. The tensor approach is then challenged with state-of-the-art graph algorithms to assess its performance.

To describe our methodology, section 1 presents the latest research concerning the smart contracts and the tensors. Section 2 introduces the concepts related to the chosen tensor decomposition and the stochastic model used for interactions simulations. In section 3, a short description of the graph theory is suggested and section 4 demonstrates the experiments performed on the tensor approach and challenged with the graph analytics.

2 Related Work

Since its first definition by Nick Szabo, the security and the programming patterns of the automated protocols have been highlighted. More specifically, in [1], the authors investigate the security on smart contracts. They underline the possibility of manipulation of smart contracts within the Ethereum platform with the goal of gaining profit. They also provide a symbolic execution tool to find potential bugs within the running Ethereum smart contracts. In [2], the authors quantify smart contracts use according to their application domain as well as the common coding patterns in Ethereum. However, even with security issues raised around automated protocols, smart contracts still offer lower transactional and running costs. In [3], the authors explain how entities can make use of smart contracts to gain in competitiveness and push further innovation. In [4], the authors propose a new way of programming smart contracts with a logic-based programming. It introduces new coding behavior with the goal of further efficiency. Last but not least, the use of smart contracts and its extension towards various domains also raise the question of their legal status. In [5], the authors address the new challenges and the evolution of the law following the use of the smart contracts in the Legal Tech companies.

However, analysis of the smart contracts using tensor based approach has not yet been deeply studied among the scientific community. Different decompositions have been applied on various subjects as the survey written by Kolda and Bader in [6] exposed it. Our work concentrates on the CANDECOMP/PARAFAC (CP) decomposition which has been presented simultaneously by Harshman et al. in [7] and [8]. It has been applied a lot in neuroscience due to the image processing and its uniqueness property. In [9], Andersen and Rayens used the CP

decomposition for functional magnetic resonance images (fMRI) data analysis in three and four dimensions. Sen and Parhi also applied in [10] CP decomposition for fMRI processing with a novel algorithm for the extraction of the common task signals and spatial maps from a fMRI group as rank-1 tensors. Other authors extended the use of the CP decomposition in other domains than neuroscience: Acar et al. in [11] used for data mining, Shen et al. for Inferring network topologies in [12] and Dinç, Ertekin and Bker applied it to the quantitative resolution of related drug substances.

In this paper, we propose a tensor-based approach based on the CP decomposition on crypto-currency domain by combining the tensor decomposition to stochastic process. This new approach allows to identify and predict smart contracts interactions.

To the best of our knowledge, this is the first work of the tensor-based approach combined with the stochastic variance gamma model for analyzing and modeling smart contracts.

3 Tensor Analysis

In this section, we present the tensor theoretical background and the tensor decomposition used for the experiments. Finally, we describe the stochastic process used for the predictions of the interactions.

3.1 Tensor Description

Notation Mathematical notations and formulation follow the notations proposed by Kolda and Bader in [6]. The notations have been reused by different articles related to tensor decomposition.

Scalars are identified by lowercase letters, a. Vectors and matrices are denoted by boldface lowercase letters and boldface capital letters, respectively $\mathbf{a}$ and $\mathbf{A}$. The transpose of the matrix $\mathbf{A} \in \mathbb{R}^{I \times J}$ is denoted by A^T.

Tensor Definition Let define a N-th multidimensional array denoted by $\mathcal{X}$ such as $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times \ldots \times I_N}$. $\mathcal{X}$ is called a N-way tensor. Tensors are denoted by Euler script letters.

Tensor Operations The square root of the sum of all tensor entries squared determines the tensor norm.

$$||\mathcal{X}|| = \sqrt{\sum_{j_1=1}^{I_1} \sum_{j_2=1}^{I_2} \ldots \sum_{j_N=1}^{I_N} x_{j_1, j_2, \ldots, j_N}^2} \qquad (1)$$

The vector outer product denoted by $\circ$ between $\mathbf{u} \in \mathbb{R}^I$ and $\mathbf{v} \in \mathbb{R}^J$ results in a matrix $\mathbf{W} \in \mathbb{R}^{I \times J}$.

$$\mathbf{u} \circ \mathbf{v} = \mathbf{u}^T \mathbf{v} = \begin{bmatrix} u_1 \\ \vdots \\ u_I \end{bmatrix} \begin{bmatrix} v_1 & \ldots & v_J \end{bmatrix} = \begin{bmatrix} u_1 v_1 & u_1 v_2 & \ldots & u_1 v_J \\ \vdots & \vdots & \ldots & \vdots \\ u_I v_1 & u_I v_2 & \ldots & u_I v_J \end{bmatrix} \qquad (2)$$

A N-way tensor $\mathfrak{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times \dots \times I_N}$ is a rank one tensor if it can be written such as

$$\mathfrak{X} = \mathbf{a}^{(1)} \circ \mathbf{a}^{(2)} \circ \cdots \circ \mathbf{a}^{(N)} \tag{3}$$

The Khatri-Rao product between two matrices $\mathbf{A} \in \mathbb{R}^{I \times K}$ and $\mathbf{B} \in \mathbb{R}^{J \times K}$, denoted by $\mathbf{A} \odot \mathbf{B}$, results in a matrix $\mathbf{C}$ of size $\mathbb{R}^{IJ \times K}$. It is the column-wise Kronecker product.

$$\mathbf{C} = \mathbf{A} \odot \mathbf{B} = \begin{bmatrix} \mathbf{a}_1 \otimes \mathbf{b}_1 & \mathbf{a}_2 \otimes \mathbf{b}_2 & \cdots & \mathbf{a}_K \otimes \mathbf{b}_K \end{bmatrix} \tag{4}$$

3.2 Tensor Decomposition

For the analysis of the interactions between smart contracts, the CP decomposition is used. It has been introduced by Harshman in [7] and Carroll and Chang in [8]. It allows a tensor $\mathfrak{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ to be written as the sum of component of rank-one tensors.

$$\mathfrak{X} = \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ \cdots \circ \mathbf{a}_r^{(N)} \tag{5}$$

The minimization equation $\min_{\hat{\mathfrak{X}}} ||\mathfrak{X} - \hat{\mathfrak{X}}||$ with $\hat{\mathfrak{X}}$ the approximate tensor constructed with matrices a^i randomly initialized and $\mathfrak{X}$ the original tensor is solved with the Alternating Least Squares (ALS) method as presented by Harshman in [7] and Carroll and Chang in [8]. The matrices $\mathbf{A} = \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \in \mathbb{R}^{I \times R}$, $\mathbf{B} = \sum_{r=1}^{R} \mathbf{a}_r^{(2)} \in \mathbb{R}^{J \times R}$ and $\mathbf{C} = \sum_{r=1}^{R} \mathbf{a}_r^{(3)} \in \mathbb{R}^{I \times R}$ are successively updated according to the non-negative scheme presented by Lee and Seung in [13].

$$\begin{cases} a_{ir} \leftarrow & a_{ir} \dfrac{[\mathbf{X}_{(1)}(\mathbf{C} \odot \mathbf{B})]_{ir}}{[\mathbf{A}(\mathbf{C} \odot \mathbf{B})^T(\mathbf{C} \odot \mathbf{B})]_{ir}} \\[2ex] b_{jr} \leftarrow & b_{jr} \dfrac{[\mathbf{X}_{(2)}(\mathbf{C} \odot \mathbf{A})]_{jr}}{[\mathbf{B}(\mathbf{C} \odot \mathbf{A})^T(\mathbf{C} \odot \mathbf{A})]_{jr}} \\[2ex] c_{kr} \leftarrow & c_{kr} \dfrac{[\mathbf{X}_{(3)}(\mathbf{C} \odot \mathbf{A})]_{kr}}{[\mathbf{C}(\mathbf{B} \odot \mathbf{A})^T(\mathbf{B} \odot \mathbf{A})]_{kr}} \end{cases} \tag{6}$$

The non-negative algorithm plays a key role in the convergence of the historical calibration of the parameters of the stochastic model.

3.3 Stochastic Model For Interactions Predictions

Interactions between smart contracts can be very discontinuous among time. Some interactions appear for short period of time before vanishing, other interactions are present in all time periods while some contracts never interacts. As a consequence, a deterministic model cannot be used but a stochastic model with jumps should be preferred.

The Variance-Gamma (VG) model has been introduced by Madan, Carr and Chang in [14]. This stochastic model is a pure jump process. One of its main application is in quantitative finance as illustrated by Hull in [15]. The VG model

is defined with three parameters denoted θ, σ and $\nu \in \mathbb{R}$. θ represents the drift in the Brownian motion among time. A Brownian motion, denoted by W, is a continuous time process representing the random motion of a small particle immersed in a fluid having the same density as the particle. σ is the volatility and ν is the variance rate of the gamma time change. The drift is represented by ωt, the gamma process by h and the process to be simulated by S.

$$\begin{cases} \omega = \frac{1}{\nu} \ln(1 - \theta\nu - 0.5\sigma^2\nu) \\ h = \theta g + \sigma \sqrt{g} z \quad \text{with} \quad g \sim \Gamma(\frac{t}{\nu}, \nu), n \sim \mathcal{N}(0, 1) \\ S_t = S_0 \exp(rt + \omega t + h) \end{cases} \tag{7}$$

Based on the empirical observations, no VG drift free rate could have been identified. As a result, the parameter r is set to 0.

4 Graph Mining

In this section, we present briefly the graph analytics used as a comparison of the tensor-VG model in the experiments.

4.1 Graph Analytics

Centrality score measures the communication importance of a vertex. In this paper, degree (baseline) centrality and betweenness (flow-based) centrality have been applied. A degree centrality represents the number of vertex neighbors and gives a local view of the graph around each node. In a directed graph, each vertex has an indegree and an outdegree. Let $G = (V, E)$ $\forall T_x \in V$. In smart contracts, we can distinguish between the senders and receivers. Indegree of vertex T_x, denoted by $deg^-(T_x)$, is the number of edges which are coming into the vertex T_x. Outdegree of vertex T_x, denoted by $deg^+(T_x)$, is the number of edges which are going out from the vertex T_x. The centrality of a sender, is based on the amount of *Ether*, Ethereum crypto-currency. We measure this using the betweenness centrality measure[16], which measures how much a given vertex lies in the weighted shortest paths of other vertices. Let $\delta_{sr} = \delta_{rs}$ denote the number of shortest paths from $T_s \in V$ to $T_r \in V$ where by convention $\delta_{T_s T_s} = 1$. Let $\delta_{T_s T_r}(T_x)$ denote the number of shortest paths from T_s to T_r that some $T_x \in V$ lies on:

$$betweenness = \sum_{T_s \neq T_x \neq T_r} \frac{\delta_{T_s T_r}(T_x)}{\delta_{T_s T_r}} \tag{8}$$

A bi-clique measure allows to discover common interaction between two sets of objects in a graph. Often, bipartite graphs can be used to represent relationships across pairs of heterogeneous data [17]. A bipartite graph is partitioned into two sets of vertices which are non-empty disjoint partitions. In bipartite graph, there is no edge within the same partition. In smart contracts, an interpretation of common senders to a set of receivers member's relationships is performed by an enumeration of *maximum bicliques.*

Maximum biclique, the largest biclique, is used as an approach to group the senders and receivers members into groups to identify most common senders of a set of receivers. In the experiments, the algorithm developed in [18] has been applied.

4.2 Link Predictions

Link predictions methods are used to identify new connections in graphs. To predict newer links among two entities at a later point in time, the path based link predictions approach is applied using the *SimRank* algorithm presented in [19]. It assumes that two vertices are similar if they share the common connected vertices. Numerically, this is specified by defining a score. The set of all neighbors of vertex T_x is denoted by $\Gamma(T_x)$.

$$
\begin{cases}
score(T_x, T_x) = 1 \\
score(T_x, T_y) = \gamma \dfrac{\sum_{a \in \Gamma(T_x)} \sum_{b \in \Gamma(T_y)} score(a, b)}{|\Gamma(T_x)|.|\Gamma(T_y)|}
\end{cases}
\quad \text{with} \quad \gamma = [0, 1] \quad (9)
$$

Finally, the SimRank corresponds to the link predictions in a bi-clique.

5 Experiments and Discussions

This section presents the experiments and the results of the tensor and the graph based approaches on smart contract data.

5.1 Dataset Characteristics

Smart contracts data have been collected from the Ethereum platform from 7 August 2015 to 2 March 2016. In this period of time, two millions of transactions have been realized between 241,385 sender accounts and 359,798 receiver accounts. A sender account is defined as an account sending Ether, the Ethereum crypto-currency, and a receiver account as an account receiving Ether. However, 60% of the sender contracts only send one Ether payment from August to March and 70% of the receiver contracts only receive once. We concentrate on contracts having the most activities, and thus we keep only the 1% most active contracts over time. In 1% of contracts, the average of contracts by sender is 46.91 and by receiver is 26.06. In our experiments, 459 smart contracts senders are considered as well as 813 smart contracts receivers.

5.2 Tensor Decomposition And Stochastic Simulation For Interactions Predictions

Tensor Decomposition Applied to Smart Contracts A three-way topological tensor is built using the data from Ethereum. A value of zero means no Ether were exchanged between a sender account and a receiver account at a

given time, and a value of one corresponds to the opposite. The first dimension of the tensor, denoted by I, characterizes the list of the sender accounts, the second dimension, J, the list of the receiver accounts and the third dimension, K, the time. Ether exchanges have been gathered in fifty-two time intervals. The size of the resulting tensor is $459 \times 813 \times 52$.
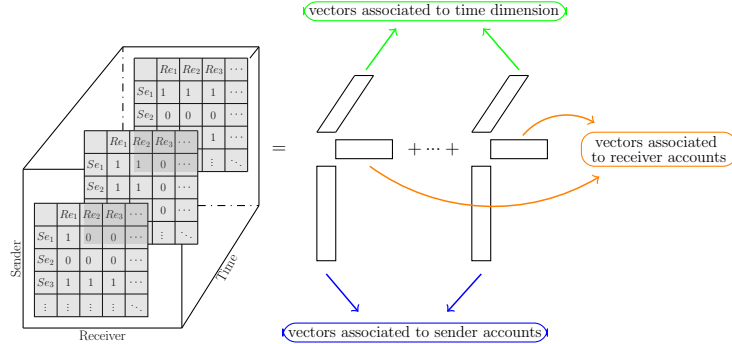

Fig. 1: Description of the tensor dimensions and interactions between sender and receiver contracts

Evaluation Of Interactions Probabilities Using VG Model The VG model is calibrated historically with the Maximum Likelihood Estimator (MLE) as presented in [20] based on the results of the tensor decomposition as described in Fig. 1. The first twenty-six time events are used for calibration and the simulations are done on the next twenty-six time events for a comparison between the true data and the simulated data.

The future probabilities of Ether exchanges at maturity T are computed with a digital function denoted by C based on the simulations of the process S.

$$C_T = \frac{1}{N} \sum_{n=1}^{N} \mathbb{1}_{S_T^{(n)} \geq K} \tag{10}$$

At maturity T, if the simulated series is equal or higher than K, then the value of C is equal to one, otherwise the value of C is equal to 0. In our simulation, we define $K = 0.99$ due to the numerical error from the tensor decomposition and the calibration of the stochastic model. The number of Monte-Carlo simulations, denoted by N, is equal to one million.

Simulation of Interaction Probabilities Using CP and VG We define a threshold of 60% as the threshold identifying an interaction or not. At the end of the time horizon of the simulation, if an interaction probability between one sender and one receiver is below the threshold, it is more likely that no interaction happen. On the opposite, if an interaction probability is higher than the threshold, it is more likely that an interaction happen.

Regarding Fig. 2a, interaction probabilities decrease over time for a given sender and a given receiver. At the end of the simulation, the interaction probability is around 55%. Given the threshold definition, it is more likely no interactions happen. The results are cross-validated with the true data that confirm no interaction happened.
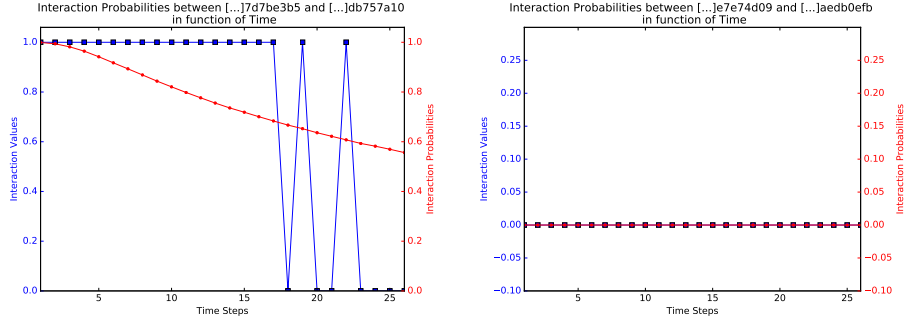
(a) Interaction Probabilities between ...7d7be3b5 and ...db757a10.

(b) Interaction Probabilities between ...e7e74d09 and ...aedb0efb.

Fig. 2: Simulation of Interactions Predictions For 26 Time Events.

Similarly, for another sender and receiver extracted from the CP decomposition in Fig. 2b, the simulation indicates that the interaction probability is 0. It is also confirmed by the true data where no interaction have been realized.

5.3   Constructing a smart contract graph

We define the *smart contract graph* to be constructed from each transaction as a directed graph representing semantically relevant relationships between two vertex-entities which represent senders and receivers performing a transaction in a smart contract. Each vertex represents a sender s and receiver r and each edge s, r indicates that the sender s sends amount of coins to r, where $s \neq r$. The data size is reduced by the representation of intermediate vertex-entities, especially for transactions with the same senders and receivers. This is a labeled directed graph $G = (V, E, \beta)$:

- V is the set of senders and receivers;
- E is a set of edges in G. Assuming T_s and T_r, sender $s \in V$ and receiver $r \in V$, there is an edge $(T_s, T_r) \in E$ if and only if it exists a transaction between T_s and T_r.
- β is a function that assigns for each edge (T_s, T_r) the average amount of transactions l_{T_s, T_r}.

The smart contract graph represents the behavior of transactions between each vertex-entities.
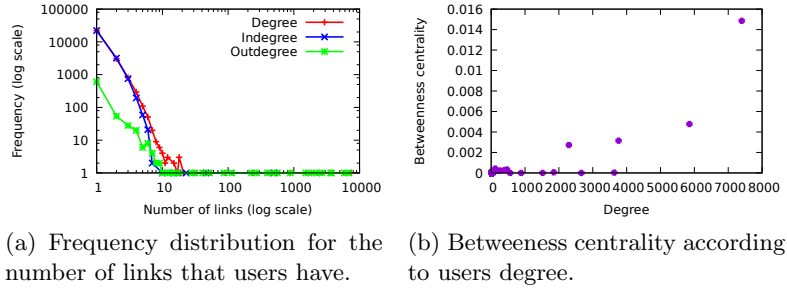
(a) Frequency distribution for the number of links that users have.

(b) Betweeness centrality according to users degree.
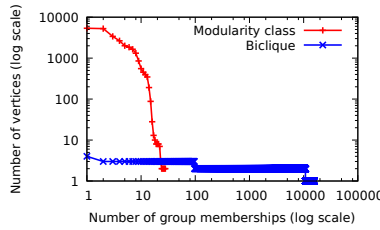
(c) Frequency distributions for group membership and group size.

Fig. 3: Main Graph Mining Results

Graph Mining Results According to Fig. 3c, the frequency of group sizes follows a highly skewed distribution. There are few and very large groups using modularity class as well as many very small bi-clique groups. 68.54% of bi-clique groups have only three members. In modularity groups, the frequency distribution for group membership is even more skewed. 10.78% of modularity groups have more than 4672 members which represents 52.95% of vertices. According to Fig. 3a, a highly skewed distribution is also observed in frequency of users' link counts. Fig. 3b highlights the most important betweenness centrality vertices. More than 51% of vertices have a degree lower than 10. The vertex degree can be higher but less important, which is equivalent to lower betweenness centrality. The betweenness centrality shows the importance of vertices or intermediate vertices during transactions. From these experiments, bi-clique groups appear more adequate to analyze smart contracts interactions by obtaining small groups leading to easier analysis.

5.4 Interactions Probabilities And Link Predictions Comparison

Four types of contracts have been tested. First, contracts having stable interactions over time or contracts having no exchange over time have been chosen. Then, two contracts, defined as type three, have been randomly selected. Type three contracts have an interaction at the initial time step and no interaction at the final time step. Finally, two contracts type four, have been selected for which

an interaction have been realized at the final time step but not at the beginning of the experiment.

Based on the first seven lines of the Table 1, the calibrated VG model is able to reproduce permanent interactions over time as well as the absence of interactions. In addition, for contracts of type three, interactions probabilities at the final time step are between 55% and 61%. It means there are very high probabilities for having no interaction. For contracts of type four, interactions probabilities at final time step are between 81% and 99.6%. As a result, the interactions probabilities determined by the VG model appear strongly correlated to the actual interactions values among the data set.

The graph probabilities could reproduce satisfyingly the absence of interactions. However, two false positives have been found with the graph approach in the illustrated sample for which no interactions have been found. One of them is linked to contracts of type four for which the interactions are the most difficult to predict. The other false positive is linked to permanent interactions which is easier to model in theory. For interactions probabilities of 29%, one is a true positive and another one is a false positive. Finally, for interactions probabilities higher than 30%, only true positives have been found.

To better compare interactions probabilities between the tensor approach and the graph approach, a curve displaying the Area Under the Curve (AUC) for Receiver Operating Characteristic (ROC) curve is shown on Fig 4.

For an interaction threshold of 20%, all interactions probabilities higher than 20% are considered as true interactions. As a result, the Fig. 4 highlights slightly better results for the graph approach for lower thresholds of interaction probabilities. However, for thresholds higher than 61% the tensor approach gives significantly better results. Overall, the tensor approach combined with the VG model appears to lead to better interactions probabilities.

Sender	Receiver	Value at $\Delta T = 0$	Value at $\Delta T = 26$	Interaction Prob. Tensor	Graph
[...]35398226	[...]d5ea2e63	1	1	1.0000	0.0000
[...]35398226	[...]a1a8170c	1	1	1.0000	0.7020
[...]01825cb5	[...]8c102d88	1	1	1.0000	0.8040
[...]b3121069	[...]9f0b1b4e	1	1	1.0000	0.4865
[...]b3121069	[...]804af9db	1	1	1.0000	0.2945
[...]2f0acb76	[...]3df1b5a3	0	0	0.0000	0.2957
[...]e7e74d09	[...]aedb0efb	0	0	0.0000	0.0000
[...]7d7be3b5	[...]db757a10	1	0	0.5587	0.0000
[...]bcda34d4	[...]c6b005d0	1	0	0.6093	0.0000
[...]35398226	[...]fdd31c8a	0	1	0.8114	0.8079
[...]49a601da	[...]41f79adc	0	1	0.9963	0.0000

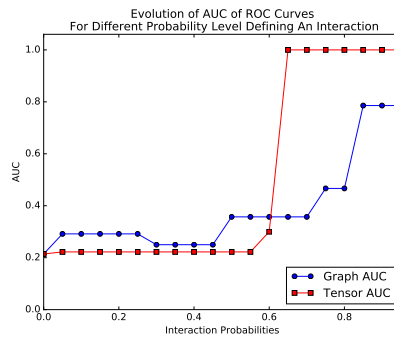Table 1: Comparison between graph and tensor based estimated interaction probability.

Fig. 4: Comparison between AUC of ROC curves for graph and tensor approaches for varying thresholds of interactions probabilities

6 Conclusion And Future Work

We proposed a tensor-based approach combined with the stochastic variance-gamma model for analyzing and modeling smart contracts operating over the Ethereum blockchain platform. The approach has been challenged with graph analysis to assess its weakness and its strengths. The tensor approach is less efficient for detecting communities in the data, but it is more appropriate for the modeling part since it can capture the interaction probabilities. The two techniques complement well each other since the weakness of one technique is the strength of the other. In addition, the accuracy of the stochastic predictions allows to identify which contract to select for speculative investment.

Our twofold future plans consist on proposing graph based link prediction on streaming smart contract and an automated process for a calibration of the stochastic parameters with lower dependency to the initial guess in the tensor approach.

Acknowledgments

References

1. Loi Luu, Duc-Hiep Chu, Hrishi Olickel, Prateek Saxena, and Aquinas Hobor. Making smart contracts smarter. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 254–269. ACM, 2016.

2. Massimo Bartoletti and Livio Pompianu. An empirical analysis of smart contracts: platforms, applications, and design patterns. *arXiv preprint arXiv:1703.06322*, 2017.

3. Vincenzo Morabito. Smart contracts and licensing. In *Business Innovation Through Blockchain*, pages 101–124. Springer, 2017.

4. Florian Idelberger, Guido Governatori, Régis Riveret, and Giovanni Sartor. Evaluation of logic-based smart contracts for blockchain systems. 2016.

5. Merit Kõlvart, Margus Poola, and Addi Rull. Smart contracts. In *The Future of Law and etechnologies*, pages 133–147. Springer, 2016.

6. Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

7. Richard A Harshman. Foundations of the parafac procedure: models and conditions for an" explanatory" multimodal factor analysis. 1970.

8. J Douglas Carroll and Jih-Jie Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of eckart-young decomposition. *Psychometrika*, 35(3):283–319, 1970.

9. Anders H Andersen and William S Rayens. Structure-seeking multilinear methods for the analysis of fmri data. *NeuroImage*, 22(2):728–739, 2004.

10. Bhaskar Sen and Keshab K Parhi. Extraction of common task signals and spatial maps from group fmri using a parafac-based tensor decomposition technique. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 1113–1117. IEEE, 2017.

11. Evrim Acar, Seyit Ahmet Camtepe, Mukkai S Krishnamoorthy, and Bülent Yener. Modeling and multiway analysis of chatroom tensors. *ISI*, 2005:256–268, 2005.

12. Yanning Shen, Brian Baingana, and Georgios B Giannakis. Inferring directed network topologies via tensor factorization. In *Signals, Systems and Computers, 2016 50th Asilomar Conference on*, pages 1739–1743. IEEE, 2016.

13. Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

14. Dilip B Madan, Peter P Carr, and Eric C Chang. The variance gamma process and option pricing. *Review of Finance*, 2(1):79–105, 1998.

15. John C Hull. *Options, futures, and other derivatives*. Pearson Education India, 2006.

16. Linton C Freeman. Centrality in social networks conceptual clarification. *Social networks*, 1(3):215–239, 1978.

17. Yun Zhang, Charles A. Phillips, Gary L. Rogers, Erich J. Baker, Elissa J. Chesler, and Michael A. Langston. On finding bicliques in bipartite graphs: a novel algorithm and its application to the integration of diverse biological data types. *BMC Bioinformatics*, 15(1):110, 2014.

18. Gabriela Alexe, Sorin Alexe, Yves Crama, Stephan Foldes, Peter L Hammer, and Bruno Simeone. Consensus algorithms for the generation of all maximal bicliques. *Discrete Applied Mathematics*, 145(1):11–21, 2004.

19. Glen Jeh and Jennifer Widom. Simrank: a measure of structural-context similarity. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 538–543. ACM, 2002.

20. Angela Loregian, Lorenzo Mercuri, and Edit Rroji. Approximation of the variance gamma model with a finite mixture of normals. *Statistics & Probability Letters*, 82(2):217–224, 2012.