Modeling Smart Contracts Activities: A Tensor based Approach

Jeremy Charlier
Security and Trust Center (SnT)
University of Luxembourg
Luxembourg, Luxembourg
Email: jeremy.charlier@uni.lu

Radu State
Security and Trust Center (SnT)
University of Luxembourg
Luxembourg, Luxembourg
Email: radu.state@uni.lu

Jean Hilger
Information Technology
BCEE
Luxembourg, Luxembourg
Email: j.hilger@bcee.lu

***Abstract*—Smart contracts are autonomous software executing predefined conditions. Two of the biggest advantages of the smart contracts are secured protocols and transaction costs reduction. On the Ethereum platform, an open-source blockchain-based platform, smart contracts implement a distributed virtual machine on the distributed ledger. To avoid denial of service attacks and monetize the services, payment transactions are executed whenever code is being executed between contracts. It is thus natural to investigate if predictive analysis is capable to forecast these interactions. We have addressed this issue and proposed an innovative application of the tensor decomposition CANDECOMP/PARAFAC to the temporal link prediction of smart contracts. We introduce a new approach leveraging stochastic processes for series predictions based on the tensor decomposition that can be used for smart contracts predictive analytics.**

***Keywords*—Tensors; CANDECOMP/PARAFAC decomposition; stochastic processes simulation**

I. Introduction

With more and more financial and IoT specific applications being implemented on top of distributed ledgers and associated monetization realized with several crypto-currencies, the modeling and predictive analytics of smart contracts is essential for multiple cases. Anti Money Laundering (AML) compliance checking is becoming mandatory and novel investment products do need a framework for modeling and analyzing smart contracts. The Ethereum platform has already more than one million accounts with little support existing in the literature on modeling and predicting the interactions among them. We have thus addressed the modeling and predictive analytics of the interactions among smart contracts from a multi-disciplinary viewpoint. We propose a multi-dimensional decomposition technique leveraging multi-dimensional tensors for extracting relevant latent factors and rely on specific time series models used in the financial industry associated to advanced calibration and Monte Carlo simulations. In order to describe our approach, we will first give a fast introduction to smart contracts and tensor models in Section 1 of the paper. Section 2 provides the fundamentals of tensor decomposition and, in Section 3, we describe the stochastic model used for the smart contracts activities prediction. We report experimental results on a large dataset in Section 4 and address a final conclusion and pointers to future works in the last section.

The main contribution of this paper consists in a tensor modeling approach for smart contracts. A second contribution is the prediction of smart contracts activities with a geometric Brownian motion combined with a Ornstein-Uhlenbeck process.

A. Smart Contracts Background

The computer scientist, Nick Szabo, introduced in 1994 the expression smart contracts 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. Related economic goals include lowering [...] transaction costs*". Smart contracts have found a direct application in the Ethereum platform that allows every programmer to create their smart contracts to send crypto token. Ethereum claimed transparent transaction and execution through a democratic organization which ensures more stability than a central gatekeeper. More particularly, in [1], Morabito describes how entities can leverage on smart contracts for automate transactions and cost reduction. Smart contracts are presented as an efficient way of gaining competitive advantage. Swan in [2] proposes a solution to execute smart contracts under optimal time condition linked to time specifiability. This condition is directly implemented in the code of smart contracts for automatic execution. Other evolution and programming features arrived such as logic-based programmation for smart contracts. In [3], the authors proposed logic-based algorithms for further efficiency of the logic approach applied to economic rule.

As illustrated, most of the research is currently focusing on smart contracts optimization or on the legal constraints arising with their use as done in [4] but not on their activities modeling. In our approach, we propose to focus on the analysis of the interactions between smart contracts. Moreover, using tensor decomposition and stochastic processes, the objective is to retrieve significant smart contracts activities that will be simulated over time.

B. Tensor Decomposition Applied to Smart Contracts

Tensors have appeared as a reliable technique for modeling interactions in multi-dimensional spaces after the introduction of CANDECOMP/PARAFAC (CP) decomposition by Harshman, Caroll and Chang in [5] and [6]. The ease of the results treatment is one of the main advantages of the CP decomposition. It has been widely used in different studies and has been followed by other techniques presented in the

extensive survey done by Kolda and Bader in [7]. The tensor theory can be applied from crime forecasting in New York city in [8] to international trade exchanges in [9]. The authors in [10] have showed CP decomposition offers good accuracy for time prediction when applied to noisy data. This evolution is joined by the development of tensor libraries in Python [11] as described by Kossaifi, Panagakis and Pantic. Furthermore, latest research focus on tensor scalability for their use in big data environment as shown by Kijung Shin, Lee Sael and U Kang in [12].

As illustrated by the published papers, tensors seem sufficiently versatile to be applied to smart contracts interaction analysis and forecasting activities. In addition, all the papers underline good accuracy of experiments results. However, papers have not yet proposed a method to model smart contracts interactions using a tensor approach. The CP tensor decomposition is applied on smart contracts executed on Ethereum platform which are available to all public users for transparency reasons.

II. Tensor Decomposition

In this section, we briefly describe mathematical operations involved in CP tensor decomposition before presenting the non-negative CP algorithm used for the analysis.

A. Tensor Description

Notation Terminology in this paper is very close to the one proposed by Kolda and Bader in [7] and commonly used by previous publications. Scalars are identified by lower case letters, *a*. Vectors and matrices are denoted by boldface lowercase letters and boldface capital letters, respectively **a** and **A**. High order tensors use Euler script notation as $\mathcal{X}$.

Tensor Definition Define $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times \ldots \times I_n}$ as a *n*-th multidimensional array. $\mathcal{X}$ is called a tensor of order *n*.

Tensor Operations The norm of a tensor $\mathcal{X}$ is defined as the square root of the sum of all tensor entries squared.

$$||\mathcal{X}|| = \sqrt{\sum_{j=1}^{I_1}\sum_{j=2}^{I_2}\ldots\sum_{j=n}^{I_n} x^2_{j_1,j_2,\ldots,j_n}} \tag{1}$$

The rank-*R* of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ is the number of linear components that could fit $\mathcal{X}$ exactly such that

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

with the symbol $\circ$ representing the vector outer product.

Matricization, also commonly known as unfolding or flattening, consists in the transformation of a *N*-way array into a matrix. The mode-*n* matricization of the tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$, denoted $\mathbf{X}_{(n)}$, is defined as

$$j = 1 + \sum_{\substack{k=1 \\ k \neq n}}^{N} (i_k - 1) J_k \quad \text{with} \quad J_k = \sum_{\substack{m=1 \\ m \neq n}}^{k-1} I_m \tag{3}$$

where the $(i_1, ..., i_d)$ tensor element is mapped to (i_n, j) matrix element.

The *n*-mode product of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ with a matrix $\mathbf{M} \in \mathbb{R}^{J \times I_n}$, denoted $\mathcal{X} \times_n \mathbf{M}$ results in a tensor of size $I_1 \times \ldots \times I_{n-1} \times J \times I_{n+1} \times \ldots \times I_N$. The *n*-mode product is defined by the following equation:

$$(\mathcal{X} \times_n \mathbf{M})_{i_1 \ldots i_{n-1} j i_{n+1} i_N} = \sum_{i_n=1}^{I_n} x_{i_1 i_2 \ldots i_n} m_{j i_n} \tag{4}$$

The Kronecker product between two matrices $\mathbf{A} \in \mathbb{R}^{I \times J}$ and $\mathbf{B} \in \mathbb{R}^{K \times L}$, denoted by $\mathbf{A} \otimes \mathbf{B}$, results in a matrix $\mathbf{C} \in \mathbb{R}^{IK \times KL}$.

$$\mathbf{C} = \mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \cdots & a_{1J}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \cdots & a_{2J}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1}\mathbf{B} & a_{I2}\mathbf{B} & \cdots & a_{IJ}\mathbf{B} \end{bmatrix} \tag{5}$$

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.

$$\mathbf{C} = \mathbf{A} \odot \mathbf{B} = [\mathbf{a}_1 \otimes \mathbf{b}_1 \quad \mathbf{a}_2 \otimes \mathbf{b}_2 \quad \cdots \quad \mathbf{a}_K \otimes \mathbf{b}_K] \tag{6}$$

B. Tensor Decomposition

In our approach, we use the CP decomposition introduced by Harshman in [5] and Carroll and Chang in [6]. This decomposition has the advantage of being one of the simplest tensor decomposition (see Fig. 1). It represents a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ as the sum of component of vector outer products.

$$\mathcal{X} = \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ \cdots \circ \mathbf{a}_r^{(N)} \tag{7}$$

To achieve the computation of the CP decomposition, the following minimization equation has to be solved.

$$\min_{\hat{\mathcal{X}}} ||\mathcal{X} - \hat{\mathcal{X}}|| \tag{8}$$

with $\hat{\mathcal{X}}$ the approximate tensor described by the CP decomposition and $\mathcal{X}$ the original tensor.

To solve 8), the Alternating Least Squares (ALS) method is used as presented by Harshman in [5] and Carroll and

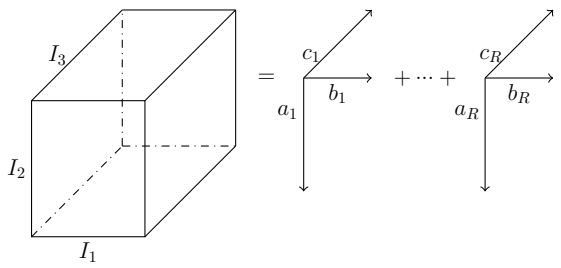

Fig. 1. CANDECOMP/PARAFAC decomposition into R components of a three way tensor.

Chang in [6]. In the experiments, we use the non-negative CP decomposition introduced by Lee and Seung in [13] for easier post-treatment. The matrices $\mathbf{A} \in \mathbb{R}^{I \times R}$, $\mathbf{B} \in \mathbb{R}^{J \times R}$ and $\mathbf{C} \in \mathbb{R}^{K \times R}$ are now updated according to the multiplicative update rule for a tensor of size $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$.

$$\begin{cases} a_{ir} \leftarrow & a_{ir} \dfrac{[\mathbf{X}_{(1)}(\mathbf{C} \odot \mathbf{B})]_{ir}}{[\mathbf{A}(\mathbf{C} \odot \mathbf{B})^T(\mathbf{C} \odot \mathbf{B})]_{ir}} \\ b_{jr} \leftarrow & b_{jr} \dfrac{[\mathbf{X}_{(2)}(\mathbf{C} \odot \mathbf{A})]_{jr}}{[\mathbf{B}(\mathbf{C} \odot \mathbf{A})^T(\mathbf{C} \odot \mathbf{A})]_{jr}} \\ c_{kr} \leftarrow & c_{kr} \dfrac{[\mathbf{X}_{(3)}(\mathbf{C} \odot \mathbf{A})]_{kr}}{[\mathbf{C}(\mathbf{B} \odot \mathbf{A})^T(\mathbf{B} \odot \mathbf{A})]_{kr}} \end{cases} \tag{9}$$

The multiplicative update rule helps to better calibration of the stochastic processes that uses the components of the tensor decomposition as a starting point.

III. Stochastic Series Prediction

In this section, we present first separately log-normal and mean-reverting stochastic models and then, we propose our approach consisting in a log-normal-mean-reverting stochastic model used for series prediction on smart contracts activities.

A. Log-Normal Stochastic Diffusion Process

The log-normal stochastic diffusion model, also known as geometric Brownian motion, is a continuous-time stochastic process. It is the solution of one of the most popular model in finance, the Black-Scholes model, introduced by Black and Scholes in [14].

The model describes the evolution of a stock which is supposed to have a log-normal distribution of its returns. The stochastic process S with a constant drift $\mu \in \mathbb{R}$, a constant volatility $\sigma \in \mathbb{R}$ and a Wiener process W follows a geometric Brownian motion if the following equation is satisfied:

$$dS_t = S_t(\mu dt + \sigma dW_t) \tag{10}$$

The Wiener process, or Brownian motion, denoted by W was introduced by R. Brown in [15] and represents the random motion of a small particle immersed in a fluid with the same density as the particle.

B. Mean Reverting Stochastic Diffusion Process

A mean-reverting process, also known as Ornstein-Uhlenbeck process, is a stochastic process that describes the velocity of a Brownian particle under friction. The process tends to evolve towards a specific long-term mean and it has been introduced by Ornstein and Uhlenbeck in [16]. This process was also generalized by Vasicek in [17] for wider application, especially in finance.

The stochastic process r with a mean reversion speed $\lambda \in \mathbb{R}$, a long term mean $\kappa \in \mathbb{R}$, a volatility $\sigma \in \mathbb{R}$ and a Wiener process W satisfies the following stochastic differential equation:

$$dr_t = \lambda(\kappa - r_t)dt + \sigma dW_t \tag{11}$$

C. Log-Normal-Mean-Reverting Model

Our approach for the series modeling consists in the use of both the Ornstein-Uhlenbeck process and the geometric Brownian motion. The rationale is if a time series follows a log-normal distribution, it could be modeled according to the geometric Brownian motion model. On one side, volatility could be calibrated on the past evolution of the time series. On the other side, the drift should represent long term behavior if there is no volatility in the data set. In our log-normal-mean-reverting model, the drift is modeled with the Ornstein-Uhlenbeck process. Let define S as the stochastic series process, μ as the stochastic drift process, $\sigma^{(S)}, \sigma^{(\mu)} \in \mathbb{R}$ the series volatility and the drift volatility, $\lambda, \kappa \in \mathbb{R}$ the mean-reversion speed and the long term mean, W as a Brownian motion and ρ as the correlation. Our model is defined by the system of equations below:

$$\begin{cases} dS_t = S_t(\mu_t dt + \sigma^{(S)} dW_t^{(1)}) \\ d\mu_t = \lambda(\kappa - \mu_t)dt + \sigma^{(\mu)} dW_t^{(2)} \\ \rho dt = d\langle W^{(1)}, W^{(2)} \rangle_t \end{cases} \tag{12}$$

The correlation denoted by ρ characterizes the correlation between the two Brownian motions of the Geometric Brownian Motion and the Ornstein-Uhlenbeck process, denoted respectively by $W^{(1)}$ and $W^{(2)}$.

IV. Experiments

In this section, we describe the data used for the tensor decomposition and the simulation of smart contracts activities using our log-normal-mean-reverting model with the goal of speculative investment.

All the experiments are performed on a PC with Intel Core i7 CPU and 8 GB of RAM. The algorithm for non-negative CP decomposition and stochastic processes has been implemented in Python language.

A. Data from Smart Contracts and Tensor Completion

Smart contracts data have been collected from the Ethereum platform starting from 7 August 2015 and ending on 2 March 2016. Different fields have been gathered such as hash key, sender accounts, receiver accounts, amount of Ether exchanged per transaction between two accounts and block heights. For the period considered within the data set, two millions transactions have been recorded. The average amount per transaction is approximately 76 Ethers. The average number of transactions per sender account is 47 transactions and per receiver account is 26 transactions.

A three-way tensor is defined according to the smart contracts data. The first dimension of the tensor, I, represents the sender accounts, the second dimension of the tensor, J, the receiver accounts and the third dimension, K, the time slot. The interaction at a given time slot between a sender account and a receiver account is represented by the amount of Ether exchanged.

B. Selection of the Smart Contracts Data for Tensor Decomposition

Among the data collected from the Ethereum platform, around 60% of the sender contracts send only one payment (Fig. 2). That is around 25,000 contracts (Fig. 3). Around 70% of the contracts, 50,000 contracts, receive only one payment for the time period considered (Fig. 4). To concentrate more on regular smart contract activities, we decide to consider the 1% most active contracts during the time. The resulting tensor has a size of 459×813×52.

C. Application of the Non-Negative CP Decomposition

Non-negative CP decomposition is applied to the smart contracts tensor. The choice of the use of a non-negative algorithm is mainly for easier calibration of the stochastic processes on the tensor decomposition components.

We define a stopping criterion ϵ for ALS algorithm using the evolution in the norm of the approximate tensor.

$$||\hat{\mathcal{X}}||^{k^{th}step} - ||\hat{\mathcal{X}}||^{(k-1)^{th}step} \leq \epsilon \quad , \quad \epsilon = 0.001 \qquad (13)$$

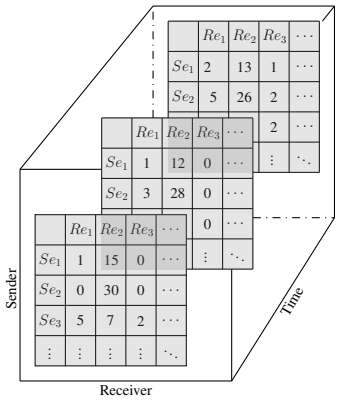

Fig. 2. Three-way tensor containing Ether amount exchanged between different sender and receiver accounts.

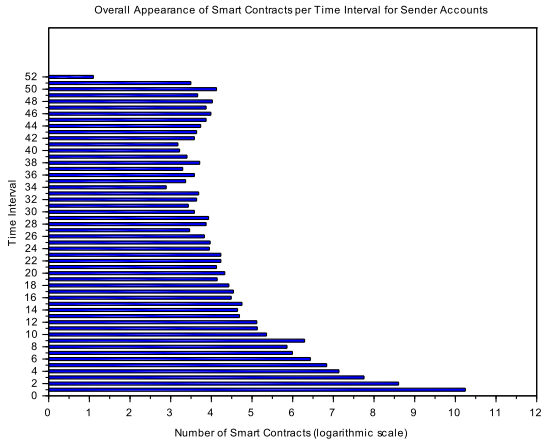

Fig. 3. More than 25,000 smart contracts only received one transaction in the time frame.

We estimate a number of rank equals to five for the tensor decomposition as the data observed within the data set can be decomposed as small exchanges, moderate exchanges, active exchanges and very active exchanges. According to the rank, the tensor decomposition highlights the interactions between senders and receivers in function of time. In Fig. 5, one sender account has been selected to visualize the Ether amount exchanged with different receiver accounts based on CP decomposition.

Furthermore, numerical experience shows that the description $\frac{\text{factor time t}}{\text{factor time t-1}}$ for a specified rank follows a log-normal distribution. To assess the accuracy of the fit to log-normal distribution, we perform the Shapiro normal test, as a distribution is said to be log-normal if the natural logarithm of the distribution is normally distributed. For our data set, we define a p-value of 10% for the null hypothesis that is the data set follows a log-normal distribution. The results are presented in Table 1.

It can be observed that the p-value of the first rank is just outside the threshold of 10%. However, we decide that the stochastic processes described in (12) would still describe properly the series for tensor rank 1 as the p-value is equal to 12.98%.

D. Use Of The Log-Normal-Mean-Reverting Process

Our time series consists in fifty-two time events. The calibration of the process S_t is performed historically using the first twenty-six time events for the simulation of the next twenty-six events, the first forty-two time events for the simulation of the next ten events and the first forty-seven time events for the simulation of the next five events. The prediction is then analyzed with the original data of the same time period to validate the approach.

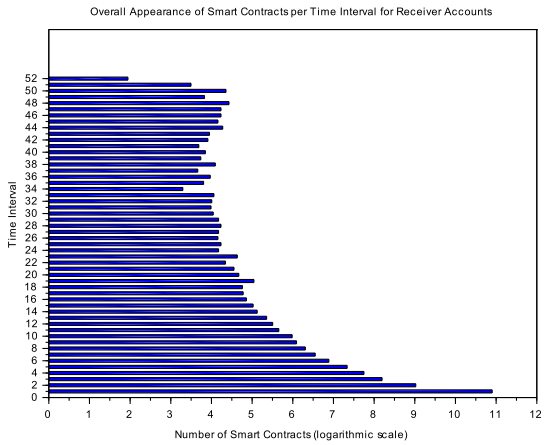

Fig. 4. More than 50,000 smart contracts only received one transaction during the time frame.

TABLE I. RESULTS OF SHAPIRO LOG-NORMAL TEST

Rank	1	2	3	4	5
pValue	0.1298	0.0003	0.0029	0.0905	0.0003

Using the system of equations described in 12, six parameters have to be calibrated: the volatility of the series $\sigma^{(S)}$, the mean reverting speed and the long term mean, λ and κ, the volatility σ^{μ} and the correlation ρ between the two stochastic processes S and μ.

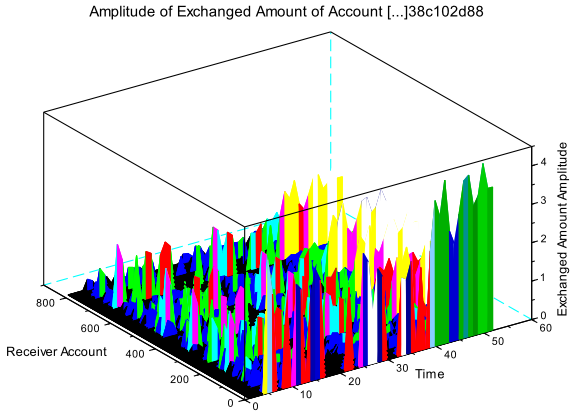

Fig. 5. Amplitude of exchanged Ether amount from a given sender to receivers during the time interval.

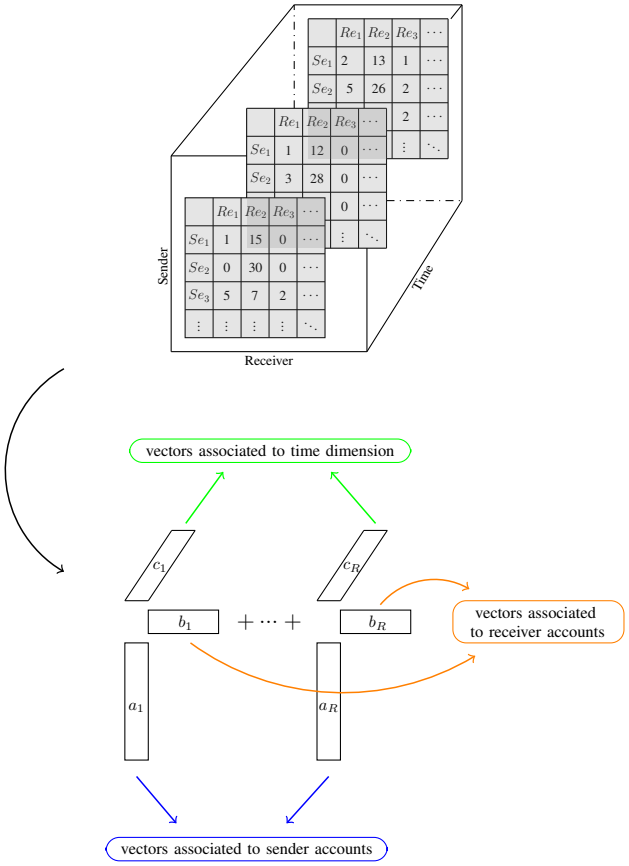

Fig. 6. Description of the dimensions related to tensor decomposition. First dimension is linked to sender accounts, second dimension to receiver accounts and third dimension to time activity. Simulations are performed on the third dimension.

The volatility of the process S_t is computed historically. The drift process μ_t illustrates the time value of money, also known as capitalization and actualization, that is one Ether today does not equal one Ether tomorrow. As a result, the parameters of the drift process μ_t are estimated on the Euro OverNight Index Average (EONIA) for the time period considered from 7 August 2015 to 2 March 2016. EONIA is the overnight rate exchanged in the interbank market. Due to the short time period of the Ether exchanged amount, it is more appropriate in this case to consider EONIA rates than other deposit rates with longer maturity. The last parameter, ρ has to be calibrated before performing the series prediction. ρ is the correlation between our time series extracted from the tensor decomposition and the EONIA historical rates. Exponential Weighted Moving Average (EWMA) correlation is used with a weight parameter of 0.9.

The values of the parameters shown in table 2 are used for the time series predictions of five time steps, ten time steps and respectively twenty-six time steps. The Monte-Carlo method is chosen to solve the system of stochastic equations presented in 12 with one million simulation.

E. Selection Of the Smart Contracts for Speculative Investment

The objective of the time series prediction using the stochastic processes is to evaluate the strength of the time vector for each tensor rank as described in Fig. 6. The selection of the smart contracts that exchange Ether is performed by assessing a probability for a time strength level.

Each of the tensor rank is associated to a particular group of smart contracts as described in subsection 4.3. Each tensor rank highlights most relevant sender contracts related to receiver contracts according to a certain time frame. A larger value of amount exchanged between a sender and a receiver is characterized by a larger value in vector time in the tensor decomposition.

For the estimation of the future probabilities of the strength of the vector time for the different tensor ranks, a digital function is applied at the maturity of the log-normal-mean-reverting stochastic process. The digital function C is defined by (14).

$$C_T = \mathbb{1}_{S_T \geq K} \tag{14}$$

If the value of the stochastic process S is below a level K at maturity T, the value of C_T is equal to 0. On the other hand, if the value of the stochastic process S is higher or equal to a level K at maturity T, the value of C_T is equal to 1. This digital description allows to estimate the probability of the process S to be higher or equal than a strike level K. The advantage of the use of the digital payoff is that the strike level can be defined according to the risk aversion of an investor. An investor having a risk averse profile would

TABLE II. RESULTS OF HISTORICAL CALIBRATION OF THE STOCHASTIC PROCESSES

Parameter	$\sigma^{(S)}$	λ	κ	σ^{μ}	ρ
5 ΔT	0.5910	0.28180	-0.0011	0.0000	-0.2621
10 ΔT	0.2010	0.2550	-0.0011	0.0000	-0.2038
26 ΔT	0.1672	0.1851	-0.0011	0.0000	-0.2288

specify a high level of strike K to maximize his probabilities of strong Ether exchanges even if it means that he might miss some opportunities. On the opposite, an investor having a risk taker profile would prefer to choose a lower strike value K even if it means that sometimes the selected contracts won't receive Ether or could even have to send lot of Ether to other smart contracts. Fig. 7, 8 and 9 illustrate the relation between the risk that an investor is ready to take according to Ether exchange probabilities. Time series have been simulated for five time steps, ten time steps and twenty-six time steps. At each time step, the value of the digital is computed to retrieve the probability of Ether exchange. The probability can be either a receiving probability if a receiver account is selected or a sending probability if a sender account is selected. Finally, the probability is compared to the actual exchange of Ether in vector time. It is important to note that the payment probability gives a confidence value on the criteria that the series will be higher than a strike level. It can be seen as a reliable indicative measure for a speculative investment according to a risk profile or an investment strategy.

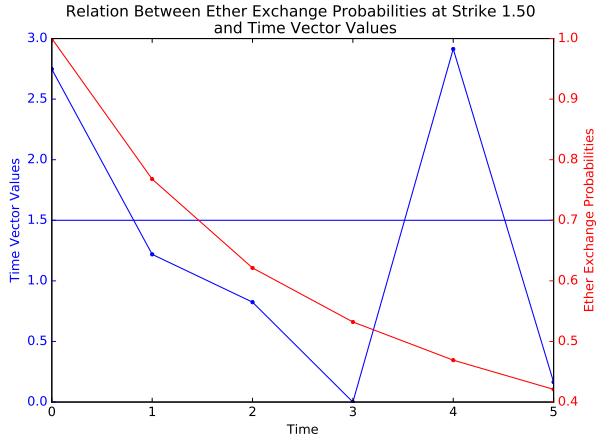

Fig. 7. Relation between the time payment magnitude and the probability of receiving cash flows over time for 5 time steps.

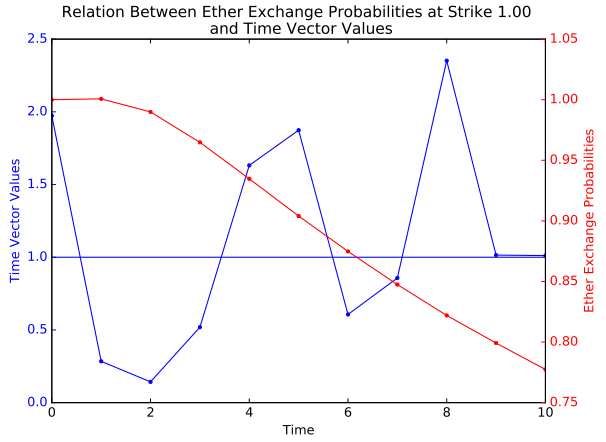

Fig. 8. Relation between the time payment magnitude and the probability of receiving cash flows over time for 10 time steps.

Tables 3, 4 and 5, the corresponding values of Fig. 7, 8 and 9, present the digital value in comparison to the actual value of the series for five time steps, ten time steps and twenty-six time steps of one tensor rank. The digital value is strongly correlated to the time series values. Simulations lose accuracy when the time step is increasing as it introduces more uncertainty with longer simulated time. Digital value below 60% reduces considerably the probabilities of exchanging Ether amount. In Table 3, the probability value of 42% means there are small probabilities of having a strong Ether exchange at fifth time step. Effectively, the time series is below the defined triggered level of 1.5. In Table 4, at the tenth time step, there is a 70% probability of exchanging Ether amount at a higher time level than 1.0. The actual value of the series confirms it with a value at the tenth time step of 1.0114. Similarly, in Table 5, at the twenty-sixth time step, there is a 0.4% probability of exchanging Ether amount at a higher strength level than 1.25 that is confirmed by the series value of 1.0114. To resume, the value of the digital can be considered as a strong indicator about the future exchanges in Ether. It provides a good source of information for speculative investment according to an investor-defined strength level of exchange in vector time.

Last but not least, the false positive and true positive rates have been calculated to determine the accuracy of the simulations (see Fig. 10). A false positive is defined when the probability of exchanging Ether is higher than 60% for a

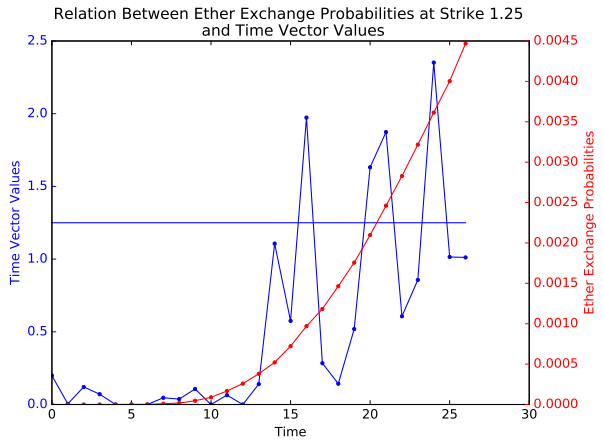

Fig. 9. Relation between the time payment magnitude and the probability of receiving cash flows over time for 26 time steps.

TABLE III. EVOLUTION OF THE DIGITAL VALUE (ETHER EXCHANGE PROBABILITY) IN RELATION WITH THE SERIES FINAL VALUE AND THE DEFINED STRENGTH LEVEL $K = 1.5$ FOR FIVE TIME STEPS SIMULATIONS

Time Step	Series Value	≥ 1.5	Digital Value
0	2.7472	-	-
5	0.1645	0	0.4218

TABLE IV. EVOLUTION OF THE DIGITAL VALUE (ETHER EXCHANGE PROBABILITY) IN RELATION WITH THE SERIES EVOLUTION AND THE DEFINED STRENGTH LEVEL $K = 1.0$ FOR TEN TIME STEPS SIMULATIONS

Time Step	Series Value	≥ 1.0	Digital Value
0	1.9732	-	-
10	1.0114	1	0.7781

strike level and no exchange of Ether happened or when the probability of exchanging Ether is below 60% for a strike level and an exchange of Ether has been realized. Similarly, a true positive is defined either when the probability of exchange Ether is higher than 60% according to a strike level and an exchange happened, or either when the probability of exchanging Ether was below the threshold of 60% according to a strike level and no Ether exchange has been observed.

V. Conclusions

We address in this paper the problem of time series prediction applied to CP tensor decomposition using a stochastic process on smart contracts. We obtain accurate probabilities prediction of Ether exchange for sender and receiver accounts that could be fitted to the risk profile of an investor or to an investment strategy. As a result, our approach can be used for the analysis of smart contract activities but also for someone who is willing to consider smart contracts as a financial investment. However, some challenges will be addressed in future work. One challenge is to use stochastic parameters for the volatility of the time series process or for the correlation involved in the stochastic equations system. It would help to increase accuracy of the simulations, in particular for longer time horizon, and to reflect deeper series variation over time. In addition, the well-known CP decomposition has been performed but other decomposition could be used to enrich the interaction analysis of the smart contracts activities such as the DEDICOM decomposition.

Acknowledgment

The authors would like to thank Beltran Borja Fiz Pontiveros for the support of Ethereum data extraction and manipulation. They also thank Jacques Putz from Banque et Caisse d'Epargne de l'Etat (BCEE) as one of the strongest support for the research publication.

TABLE V. Evolution of the Digital Value (Ether Exchange Probability) in Relation with the Series Evolution and the Defined Strength Level $K = 1.25$ for Twenty-Six Time Steps Simulations

Time Step	Series Value	≥ 1.25	Digital Value
0	0.1987	-	-
26	1.0114	0	0.0045

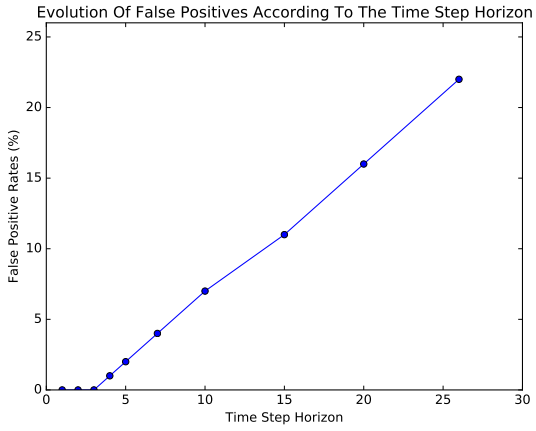

Fig. 10. Relation between the time payment magnitude and the probability of receiving cash flows over time.

TABLE VI. Evolution of the False Positive and True Positive Rates Depending on the Horizon of the Simulation for Different Strike Levels

Time Step	False Positive Rates (%)	True Positive Rates (%)
5	2	98
10	7	93
26	22	78

References

[1] Vincenzo Morabito, Smart Contracts and Licensing, Business Innovation Through Blockchain, Part II, 2017, pp. 101124, doi:10.1007/978-3-319-48478-5_6.

[2] Melanie Swan, Blockchain Temporality: Smart Contract Time Specifiability with Blocktime, Springer International Publishing Switzerland 2016, 2016, doi:10.1007/978-3-319-42019-6_12.

[3] Florian Idelberger, Guido Governatori, Rgis Riveret and Giovanni Sartor, Evaluation of Logic-Based Smart Contracts for Blockchain Systems, Springer International Publishing Switzerland 2016, 2016, doi:10.1007/978-3-319-42019-6_11.

[4] Merit Kõlvart, Margus Poola and Addi Rull, Smart Contracts, The Future of Law and eTechnologies, 2016, pp. 133-147, doi:10.1007/978-3-319-26896-5_7.

[5] R. A. Harshman, Foundations of the PARAFAC procedure: Models and conditions for an explanatory multi-modal factor analysis, UCLA Working Papers in Phonetics, vol.16, 1970, pp. 184. Available at http://publish.uwo.ca/harshman/wpppfac0.pdf

[6] D. Carroll and J. J. Chang, Analysis of individual differences in multidimensional scaling via an N-way generalization of Eckart-Young decomposition, Psychometrika, vol.35, 1970, pp. 283319.

[7] Tamara G. Kolda and Brett W. Bader, Tensor Decompositions and Applications, Society for Industrial and Applied Mathematics (SIAM) Review, vol. 62 no. 3, 2009, pp. 455-500.

[8] Tamara G. Kolda, Richard A. Harshman and Brett W. Bader, Temporal analysis of semantic graphs using ASALSAN, Sandia National Laboratories Technical Report, 2007, doi:10.1109/icdm.2007.54.

[9] Yang Mu, Wei Ding, Melissa Morabito and Dacheng Tao, Empirical Discriminative Tensor Analysis for Crime Forecasting, National Institute of Justice, 2009, no.2009-DE-BX-K219.

[10] Tamara G. Kolda, Daniel M. Dunlavy and Evrim Acar, Temporal Link Prediction using Matrix and Tensor Factorizations, Sandia National Laboratories, 2010.

[11] Jean Kossaifi, Yannis Panagakis and Maja Pantic, TensorLy: Tensor Learning in Python, 30th Conference on Neural Information Processing Systems (NIPS) Barcelona (Spain), 2016.

[12] Jean Kijung Shin, Lee Sael, and U Kang, Fully Scalable Methods for Distributed Tensor Factorization, IEEE Transactions On Knowledge And Data Engineering, 2017.

[13] D. Lee and H. Seung, Learning the parts of objects by nonnegative matrix factorization, Nature, 1999, pp. 788-791.

[14] Fischer Black and Myron Scholes, The Pricing of Options and Corporate Liabilities, Journal of Political Economy, vol. 81, 1973, pp. 637654, DOI:10.1086/260062.

[15] Robert Brown, A brief account of microscopical observations made in the months of June, July and August, 1827, on the particles contained in the pollen of plants; and on the general existence of active molecules in organic and inorganic bodies, Phil. Mag., vol. 4, 1828, pp. 161-173.

[16] George E. Uhlenbeck and Leonard S. Ornstein, On the theory of Brownian Motion, Physics Review, vol. 36, 193, pp. 823-841, DOI:10.1103.

[17] Oldrich Vasicek, An equilibrium characterization of the term structure, Journal of Financial Economics, vol. 5, 1977, pp.177188.