Law as a Service (LaaS): Enabling Legal Protection over a Blockchain Network

Muhammad Umer Wasim
Computer Science & Communications Research Unit, University of Luxembourg, Luxembourg
umer.wasim@gmail.com

Abdallah A. Z. A. Ibrahim
Computer Science & Communications Research Unit, University of Luxembourg, Luxembourg
abdallah.ibrahim@uni.lu

Pascal Bouvry
Computer Science & Communications Research Unit, University of Luxembourg, Luxembourg
pascal.bouvry@uni.lu

Tadas Limba
Digital and Creative Industries LAB, Mykolas Romeris University, Lithuania
tlimba@mruni.eu

***Abstract*—Breaches in online contracts (Service Level Agreements, SLAs) are usually compensated by gift vouchers at present, however as the online contracts emerge towards smart contracts, the breaches could potentially lead to court injunctions over blockchains. This research proposes Probability based Factor Model (PFM) that can be implemented over the blockchain to automatically identify breaches that can cause substantial damage and have high probability for recurrence. PFM can also issue court injunctions for the breaches. The underlying concept in PFM is built upon the notion of factor analysis and stochastic modeling from the discipline of Data Science. High performance computing (HPC) cluster at University of Luxembourg (HPC @ Uni.lu) and docker (a software container platform) were used to emulate contractual environment of three service providers: Redis, MongoDB, and Memcached Servers. The results showed that court injunction(s) was issued only for Redis and MongoDB Servers. Technically, this difference could be attributed to the fact that Memcached is simply used for caching and therefore, it is less prone to breach of contract. Whereas, Redis and MongoDB as databases and message brokers are performing more complex operations and are more likely to cause a breach. This research will benefit enterprises that view breach of contract as a limiting factor for implementation of smart contract in cyber-physical system or internet of things.**

Keywords— blockchain, smart contract, contract law, breach of contract, court injunction, unsupervised machine learning, factor analysis, stochastic modeling, structural equation modeling.

1. INTRODUCTION

Blockchain is an emerging technology for decentralized and transactional data sharing across a large network of untrusted participants [1]. The first generation of the blockchain was a public ledger for monetary transactions with very limited capability to support programmable transactions. The typical example is cryptocurrency or Bitcoin [2]. The second generation of the blockchain became a generally programmable infrastructure with a public ledger that records computational results. In this generation, smart contracts were introduced as autonomous programs that are deployed by the components connected to the blockchain to reach agreements and solve problems with minimal trust [3]. Autonomous Decentralized Peer-To-Peer Telemetry (ADEPT), a project of IBM is an excellent implementation of smart contracts to enable programmable transaction in cyber-physical system or internet of things [4].

A smart contract is a piece of code that resides on a blockchain and is identified by a unique address. It includes a set of executable functions and state variables. The function is executed when a transaction is invoked by a certain condition (or by an electronic event or data). These transactions include input parameters that are required by the functions in the contract, see Figure 1. Upon the execution of a function, the state variables in the contract change depending on the logic implemented in the function. This execution is self-enforceable i.e. once a smart contract is concluded, its further execution is neither dependent on intend of contractual parties or third party nor does it require any additional approvals or actions from their side [5]. Thus, any malicious intent of the party i.e. breach of contract, and role of third party addressing the malicious intent i.e. judiciary, becomes irrelevant during the execution of a smart contract [6].

Fig. 1. Smart Contract

2. LITERATURE REVIEW AND RESEARCH GAP

In addition to dealing with breaches, contract law also encompasses deviations in pre-defined outcomes. [7]. Even though breach of contract and role of judiciary become irrelevant during the execution of a smart contract, what if an output of a smart contract is considered as a breach by court of law? For example, a court may acknowledge deviation in output of a contract as a breach during litigation, e.g. 90% actual uptime of a web service instead of agreed 99% uptime.

Currently in an online contract i.e. service level agreements (SLAs), customers are commonly compensated with a gift voucher when a breach of contract occurs [8], whereas, in an emerging world of online contracts i.e. smart contracts, an automatic court injunction[1] is expected to emerge over a blockchain [9]. Current research projects that are using smart contracts as underlying technology e.g. ADEPT by IBM, Slock.it, Trans Active Grid, and Filament; have overlooked the

[1] A short order by which an entity is required to perform, or is restrained from performing, a particular act.

978-1-5386-0759-6/17/$31.00 ©2017 IEEE

need to instantiate role of judiciary over a blockchain [10]. One of the major reasons for such gap is initial level of multi-disciplinary research when it comes to provisioning legal protection over a blockchain [9]. The aim of this research is to develop a model that can be implemented over the blockchain to automatically issue court injunction for the breach, which has a potential to create substantial damage and has high probability to occur in the future. Respectively, the main research question addressed in this research is: what happens when the outcome of a smart contract deviates from the outcome that the law demands? The remaining parts of this paper are organized as follows. Section 3 gives a summary overview of the proposed model. Section 4 presents a rundown on implementation and results. Finally, section 5 concludes the paper by presenting main research findings and directions for future research.

3. PROPOSED MODEL

This research proposes an unsupervised machine learning algorithm called as Probability based Factor Model (PFM) to automatically issue a court injunction when output of a smart contract breaches the contract. The underlying concept in PFM is built upon the notion of factor analysis and stochastic modeling from the discipline of Data Science [11]. Using past data, it performs two-phase validation process to issue a court injunction. Initially, it assesses significance of a breach to ensure that the breach has a potential to create a substantial damage. Afterwards, if the significance is high, it assesses the probability of the breach. In case the probability is also high i.e. breach was frequently occurring in the past and there is certainty for it to occur in the future, PFM invokes a transaction and executes a function in a smart contract that results in the issue of court injunction. Figure 2 presents an example of a smart contract for Quality of Service (QoS) and a context when the contract is implemented with PFM.

Smart Contract for QoS	PFM based Smart Contract for QoS
Condition If latency of a cloud service goes beyond a pre-defined threshold or throughput falls below pre-defined threshold, the client machine sends a maintenance request. Transaction For sending the maintenance request, a transaction is sent to the *request_service_function* of the *Service_Smart_Contract* between the client machine and the service provider machine.	Condition (or Breach) If latency of a cloud service goes beyond a pre-defined threshold or throughput falls below pre-defined threshold, PFM at the client machine applies following logical operations to send a injunction request. φ is a high significance of the breach ϑ is a high probability of the breach **INJ** is a court injunction $(\neg\varphi \vee (\varphi \wedge \neg\vartheta) \rightarrow \neg\mathbf{INJ}) \wedge (\varphi \wedge \vartheta \rightarrow \mathbf{INJ})$ Transaction For sending the injunction request, a transaction is sent to the *request_service_function* of the *Breach_Service_Smart_Contract* between the client machine, the service provider, and the court of law.

Fig. 2. PFM enabled Smart Contract

A. Assessing Significance of Breach

To assess significance of breach, PFM uses notion of communality [11]. Communality belongs to broader concept of factor analysis from the discipline of Data Science [12, 13]. In Figure 2, it is the measure of the relationship between contract (QoS) and its output e.g. latency. Its high value indicates a strong relationship between the two and endorses the related breach e.g. latency > threshold, significant.

Communality is estimated by using structural equation modeling (SEM). SEM is a statistical approach used to examine association between a latent variable and observed variable [12, 13]. Latent variable is a theoretical construct that is inferred from the variable that is observed during the test or survey. In Figure 2, QoS (contract) is a latent variable since it represents intent of a customer and is inferred from latency or throughput (output of the contract) that is observed during the test or survey.

In SEM, the most popular and frequently used methods to estimate communality are Principal Factor Analysis (PFA) and Maximum Likelihood (ML) [12, 13]. Considering that ML estimation assumes normal distribution of observed variables and this research is dealing with observed variables without making any prior assumption, so PFA was used to estimate communality. The vector notation in PFA that is used to calculate communality (η) is given in equation 1. For summarized discussion on derivation of η see appendix.

$$\eta = \begin{bmatrix} (u_1)^2 \\ (u_2)^2 \\ \vdots \\ (u_n)^2 \end{bmatrix} \Theta_i \qquad (1)$$

In the equation, the vector contains estimated unit-scaled loadings or weights (u_i) that are associated with each observed variable. The scalar quantity Θ_i is a shared variance among all the observed variables that represent the latent variable. Communality is obtained by multiplying squared value of u_i with Θ_i, which represents the relationship of latent variable with observed variable. In Figure 2, let's say the communality obtained for "QoS and latency" and "QoS and throughput" is 0.87 and 0.14. In later case, the low value indicates weak relationship and therefore, declares the related breach i.e. throughput < threshold, insignificant and unlikely to create substantial damage.

For comparative analysis, η is compared with the results of competing research models from the domain of Multi-Criteria Decision Analysis (MCDA): Analytic Hierarchy Process (AHP) and Technique for Order Preference by Similarity to an Ideal Solution (TOPSIS)[14].

B. Assessing Probability of Breach

To assess probability of breach $P(x)$, PFM uses notion of stochastic modeling. A stochastic model predicts a random event weighted by its probability [15]. PFM, based on the distribution modeling of the previous breaches $(x_{t-1}, x_{t-2}, .., x_{t-n})$, suggests a stochastic model with minimum "square error" to find $P(x)$. In distribution modeling, square error as criteria with the minimum value indicates best possible approximation (stochastic model) for the data. However, the best possible approximation also requires verification in terms of accuracy i.e. how precisely a stochastic model can represent the data.

For example, during the distribution analysis, if PFM observes previous beaches are lognormal increasing with minimum square error, then the stochastic model in equation 2 will be used by PFM to calculate probability of breach P(x).

$$P(x) = \left\{ \frac{1}{\sigma x \sqrt{2\pi}} e^{-(ln(x)-\mu)^2/(2\sigma^2)} \;\; if \;\; (x_{t-1}, .., x_{t-n}) \sim LOGN(\mu, \sigma) \right. \qquad (2)$$

To verify the accuracy of above model, PFM performs a Paired Sample T-Test. In the test, it determines whether the mean difference between two samples i.e., previous breaches and random data generated using $LOGN(\mu, \sigma)$ in equation 2, is

zero or not. For later case i.e. ≠ 0, PFM dismisses the use of stochastic model in equation 2.

4. Evaluation and Results

High performance computing (HPC) cluster at University of Luxembourg (HPC @ Uni.lu) and docker (a software container platform) were used to emulate contractual environment of three service providers: Redis, MongoDB, and Memcached Servers. Each of these service providers were operating under a workload comprising of different number of operations ranging from 0 to 10,000, number of records ranging from 0 to 10,000, and number of threads ranging from 0 to 100.

Yahoo Cloud Service Benchmark (YCSB) was deployed at the customer machine, to continuously monitor QoS of service providers in terms of throughput (operations per second), read latency (time to read data from database), and update latency (time to update data in database).

The breach of contract was emulated by increasing the workload to influence throughput, read latency, and update latency of service providers. Python (for scripting) and R/R Studio (for data visualization) were used to identify the breach and consequently, PFM was activated to issue a court injunction. The data analysis tools that assisted PFM were: Arena Input analyzer, STATA, IBM Statistical Analysis Software Package (SPSS), and Microsoft Excel.

Figure 3 presents YCSB monitoring of service providers in terms of unit-scaled throughput, read latency, and update latency. The YCSB data of all three service providers was used by PFM to calculate communality for throughput (0.38), read latency (0.46), and update latency (0.33). It can be observed that read latency has highest value and consequently, the strongest relationship with QoS. Therefore, the related breach i.e. read latency > threshold, is significant and most likely to create substantial damage. For comparative analysis, communality for throughput (0.38), read latency (0.46), and update latency (0.33) was compared with the results of AHP and TOPSIS. The results show better performance of PFM as compared to AHP and TOPSIS. Because of space limitation, the detail of comparative analysis is not presented in the paper.

For each service provider, (a) the threshold was set to average read latency, which was calculated from its YCSB data, (b) based on the condition i.e. read latency > average read latency, previous breaches $(x_{t-1}, x_{t-2}, .., x_{t-n})$ were identified, (c) distribution modeling of previous breaches was performed using PFM, (d) afterwards, stochastic model with minimum square error was identified, and further verified for accuracy using Paired Sample T-Test.

The stochastic models for read latency of Redis and Memcached successfully passed the T-Test. However, for MongoDB (as it failed the prior T-Test) the procedure in preceding paragraph was repeated for throughput (with second highest communality value of 0.38) and stochastic model identified successfully passed the T-Test.

Table 1 presents the implementation and results of PFM. Row 1 of the table shows previous breaches based on two conditions: "read latency > average read latency" for Redis and Memcached, and "throughput < average throughput" for MongoDB. Row 2 of the table shows distribution modeling results. It can be observed that for Redis and Memecached, previous breaches in read latency are lognormal increasing and for MongoDB, previous breaches in throughput are beta increasing.

Row 3 of the table presents stochastic models for each service provider with minimum square error (Redis: 0.007417, Memcashed: 0.003444, and MongoDB: 0.018634). Moreover, as p-values of Paired Sample T-Test (Redis: 0.5449, Memcashed: 0.8258, and MongoDB: 0.4788) are greater than 0.05, the null hypothesis (the two samples are same) is accepted as compared to alternate hypothesis (the two samples are different). Hence, the stochastic models for Redis (read latency) i.e., 0.12 + LOGN(0.204, 0.117), Memcached (read latency) i.e., 0.27 + LOGN(0.245, 0.137), and MongoDB (throughput) i.e-0.48 + 0.17 * BETA(2.49, 1.48), can be used by PFM to find probability of breach $P(x)$.

Last row in table 1 shows lognormal $P(x)$ for Redis and Memcached and beta $P(x)$ for MongoDB. Last row in table 1 also shows issued injunctions. Based on the opinion of substantive specialist in the field and communality, for Redis and Memcached the injunction was issued based on the condition: $P(x) > 0.70$, whereas, for MongoDB the condition was: $P(x) > 0.45$. It can be observed that court injunction(s) was issued only for Redis and MongoDB Servers. Technically, this difference could be attributed to the fact that Memcached is simply used for caching and therefore, it is less prone to breach of contract. Whereas, Redis and MongoBD as databases and message brokers are performing more complex operations and are more likely to cause a breach.

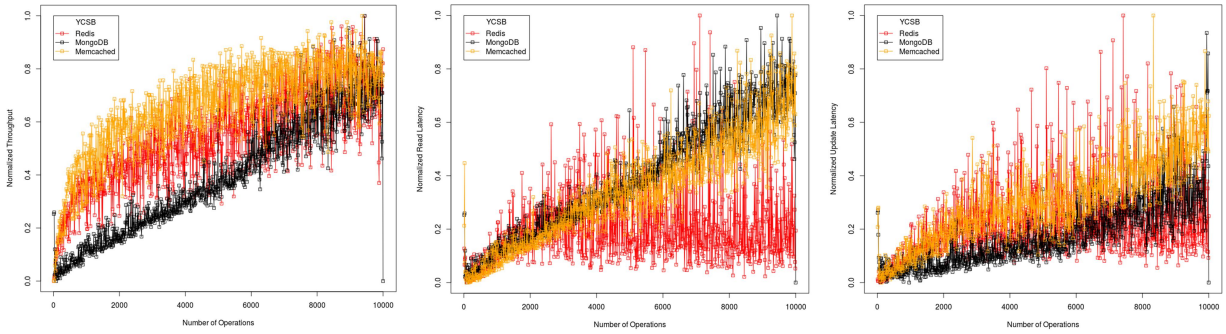
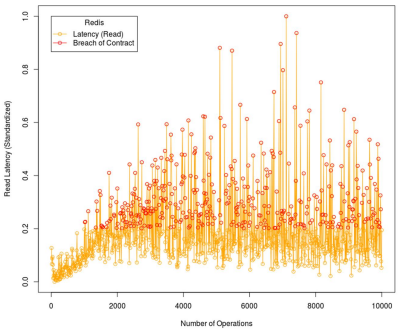
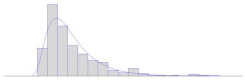


Fig. 3. YCSB (version 0.12.0) Monitoring of Redis, MongoDB, and Memcached

TABLE I. IMPLEMETATION AND RESULTS OF PROBABILITY BASED FACTOR MODEL (PFM)

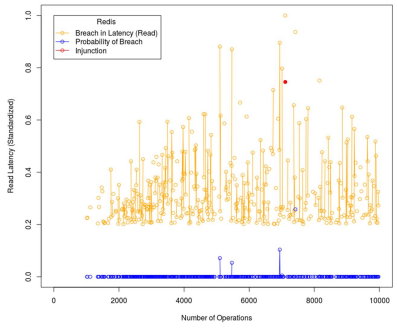


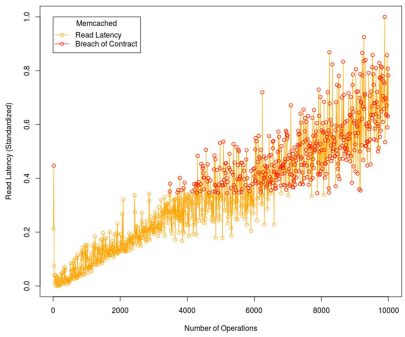


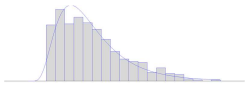
Distribution: Lognormal

Stochastic Model: 0.12 + LOGN(0.204, 0.117)
where,
LOGN(LogMean μ, LogStd σ)
LogMean μ = 0.204
LogStd σ = 0.117
Offset = 0.12
Square Error: 0.007417
p-value (t-test): 0.5449 (>0.05)
Equation:

$$P(x) = \frac{1}{\sigma x\sqrt{2\pi}} e^{-(ln(x)-\mu)^2/(2\sigma^2)}$$



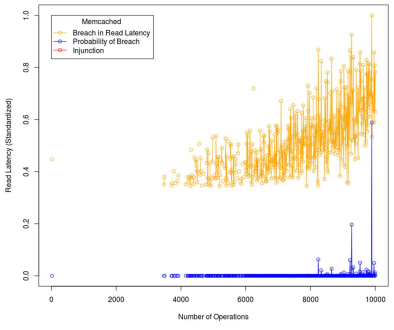


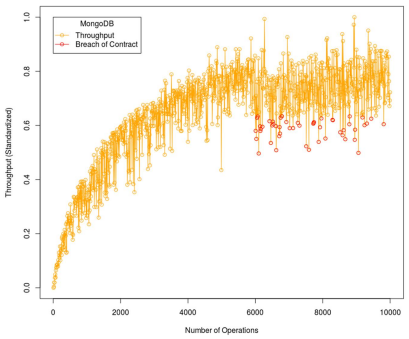


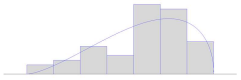
Distribution: Lognormal

Stochastic Model: 0.27 + LOGN(0.245, 0.137)
where,
LOGN(LogMean μ, LogStd σ)
LogMean μ = 0.245
LogStd σ = 0.137
Offset = 0.27
Square Error: 0.003444
p-value (t-test): 0.8258 (>0.05)
Equation:

$$P(x) = \frac{1}{\sigma x\sqrt{2\pi}} e^{-(ln(x)-\mu)^2/(2\sigma^2)}$$



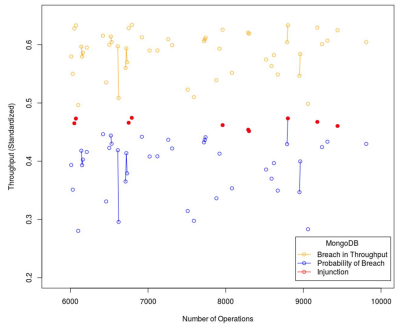




Distribution: Beta

Stochastic Model: 0.48 + 0.17 * BETA(2.49, 1.48)
where,
BETA(Beta β, Alpha α) or *BETA(Alpha1, Alpha2)*
Beta β (Alpha1) = 2.49
Alpha α (Alpha2) = 1.48
Offset = 0.48 + (0.17 * BETA)
Square Error: 0.018634
p-value (t-test): 0.4788 (>0.05)
Equation:

$$P(x) = \frac{x^{\beta-1}(1-x)^{\alpha-1}}{\int_0^1 t^{\beta-1}(1-t)^{\alpha-1}dt}$$



5. CONCLUSION AND FUTURE RESEARCH

This research proposes Probability based Factor Model (PFM) that can be implemented over the blockchain to automatically issue court injunction for the breach of contract, which has a potential to create substantial damage and has high probability to occur in the future. The underlying concept in PFM is built upon the notion of factor loading and stochastic modeling from the discipline of Data Science. High performance computing (HPC) cluster at University of Luxembourg (HPC @ Uni.lu) and docker (a software container platform) were used to emulate contractual environment of three service providers: Redis, MongoDB, and Memcached Servers. The breach of contract was emulated by increasing the workload on these providers. The results showed that the court injunction(s) was issued only for Redis and MongoDB Servers. Technically, this difference could be attributed to the fact that Memcached is simply used for caching and therefore, it is less prone to breach of contract. Whereas, Redis and MongoDB as databases and message brokers are performing more complex operations and are more likely to cause a breach. Moreover, the results of MongoDB server show the limitation of PFM when stochastic model fails the T-Test. In the next stage of the research, the goal is to test PFM in real time blockchain environment.

APPENDIX

In PFA, the relationship vector $\Lambda = (\lambda_1\lambda_2 \ldots \lambda_n)'$ between a latent variable F and observed variable vector $Y = (y_1 y_2 \ldots y_n)'$ is expressed in a variance-covariance matrix notation as:

$$\text{cov}(Y) = \text{cov}(\Lambda F) + \psi$$

ψ is a vector that represent uniqueness of observed variables not shared with the latent variable. By using covariance property $\text{cov}(AZ) = A\,\text{cov}(Z)\,A^T$, $\text{cov}(\Lambda F)$ in the right hand

side of above equation can be expanded to $\Lambda\,\mathrm{cov}(F)\,\Lambda^T + \psi$. Moreover, since F being an identity matrix has $\mathrm{cov}(F) = 1$, $\Lambda\,\mathrm{cov}(F)\,\Lambda^T$ can be further reduced to: $\Lambda\Lambda^T + \psi$ and the equation becomes:

$$\mathrm{cov}(Y) = \Lambda\Lambda^T + \psi$$

If Y is not commensurate i.e. observed variables are measured in different units and scales, then standardized Y is used. After standardization, covariance becomes correlation (r) and subsequently, covariance matrix $\mathrm{cov}(Y)$ becomes a correlation matrix R.

$$R = \Lambda\Lambda^T + \psi$$

we can expand above equation as:

$$\begin{bmatrix} 1 & \cdots & r_{1n} \\ \vdots & \ddots & \vdots \\ r_{n1} & \cdots & 1 \end{bmatrix} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \end{bmatrix} [\lambda_1\ \lambda_2 \dots \lambda_n] + \begin{bmatrix} \psi_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \psi_n \end{bmatrix}$$

Bringing ψ to left hand side and preforming subtraction,

$$\begin{bmatrix} 1-\psi_1 & \cdots & r_{1n} \\ \vdots & \ddots & \vdots \\ r_{n1} & \cdots & 1-\psi_n \end{bmatrix} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \end{bmatrix} [\lambda_1\ \lambda_2 \dots \lambda_n]$$

Subtracting unique variance from the one $(1-\psi_i)$ will yield shared variance of an observed variable for the latent variable, which is equal to square of λ_i. Respectively, $(\lambda_i)^2$ can replace $1-\psi_i$ and above equation will become:

$$\begin{bmatrix} (\lambda_1)^2 & \cdots & r_{1n} \\ \vdots & \ddots & \vdots \\ r_{n1} & \cdots & (\lambda_n)^2 \end{bmatrix} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \end{bmatrix} [\lambda_1\ \lambda_2 \dots \lambda_n] \quad (1)$$

Where left hand side,

$$\begin{bmatrix} (\lambda_1)^2 & \cdots & r_{1n} \\ \vdots & \ddots & \vdots \\ r_{n1} & \cdots & (\lambda_n)^2 \end{bmatrix} = R - \psi$$

Accordingly, in a reduce form, equation 1 becomes:

$$R - \psi = \Lambda\Lambda^T \quad (2)$$

$R-\psi$ is a 'reduced correlation matrix' with $(\lambda_i)^2$ on the diagonal. If $R-\psi$ is positive semi-definite matrix i.e. it satisfy $R-\psi = (R-\psi)^T$, then this implies that left hand side in equation 2 is symmetric and has a following spectral decomposition.

$$R - \psi = UDU^T \quad (3)$$

Spectral decomposition is the factorization of a matrix into a canonical form, whereby the matrix is represented in terms of its eigenvectors to identify latent variable and corresponding eigenvalues to show strength of identified latent variable. In equation 3, U is the matrix of eigenvectors of $R-\psi$ and D is the diagonal matrix of corresponding eigenvalues $\Theta_1\ \Theta_2 \dots \Theta_n$.

$$D = \begin{bmatrix} \Theta_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \Theta_n \end{bmatrix}$$

The important property of a positive semi-definite matrix is that its eigenvalues are always positive or null. Hence, $\Theta_i \geq 0$ and consequently, D can be factored into $D^{1/2}D^{1/2}$ and right hand side in equation 3 becomes:

$$R - \psi = \left(UD^{\frac{1}{2}}\right)\left(D^{\frac{1}{2}}U^T\right) \quad (4)$$

Equation 4 is in the form of equation 2 and accordingly, following can be deduced for Λ.

$$\Lambda = \left(UD^{\frac{1}{2}}\right)$$

In an expanded form, right hand side in above equation can be written as:

$$\Lambda = \begin{bmatrix} u_{11} & \cdots & u_{1n} \\ \vdots & \ddots & \vdots \\ u_{n1} & \cdots & u_{nn} \end{bmatrix} \times \begin{bmatrix} \sqrt{\Theta}_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sqrt{\Theta}_n \end{bmatrix}$$

It can be observed that Λ (or $UD^{1/2}$) is $[n \times n]$ matrix, however, for single latent variable F, Λ must be $[n \times 1]$ matrix as $\Lambda = (\lambda_1\lambda_2 \dots \lambda_n)'$. Hence, from the right hand side of above equation we take the largest eigenvalue Θ_i and corresponding eigenvector U_i for calculation of Λ i.e., $\Lambda = U_i\sqrt{\Theta_i}$. Whereas, using Λ, communality η is calculated as:

$$\eta = \Lambda^2 = \begin{bmatrix} (u_1)^2 \\ (u_2)^2 \\ \vdots \\ (u_n)^2 \end{bmatrix} \Theta_i$$

ACKNOWLEDGEMENT

Research presented in this paper is conducted as a PhD research at the University of Luxembourg, within the Erasmus Mundus Joint International Doctoral (Ph.D.) program in Law, Science and Technology.

REFERENCES

[1] M. Swan, Blockchain: Blueprint for a new economy: " O'Reilly Media, Inc.", 2015.

[2] S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G. M. Voelker, et al., "A fistful of bitcoins: characterizing payments among men with no names," in Proceedings of the 2013 conference on Internet measurement conference, 2013, pp. 127-140.

[3] K. Delmolino, M. Arnett, A. Kosba, A. Miller, and E. Shi, "Step by step towards creating a safe smart contract: Lessons and insights from a cryptocurrency lab," in International Conference on Financial Cryptography and Data Security, 2016, pp. 79-94.

[4] X. Xu, C. Pautasso, L. Zhu, V. Gramoli, A. Ponomarev, A. B. Tran, et al., "The blockchain as a software connector," in Software Architecture (WICSA), 2016 13th Working IEEE/IFIP Conference on, 2016, pp. 182-191.

[5] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," IEEE Access, vol. 4, pp. 2292-2303, 2016.

[6] A. Savelyev, "Contract law 2.0:'Smart'contracts as the beginning of the end of classic contract law," Information & Communications Technology Law, vol. 26, pp. 116-134, 2017.

[7] C. L. Knapp, N. M. Crystal, and H. G. Prince, Problems in Contract Law: cases and materials: Wolters Kluwer Law & Business, 2016.

[8] F. Campanile, L. Coppolino, S. Giordano, and L. Romano, "A business process monitor for a mobile phone recharging system," Journal of Systems Architecture, vol. 54, pp. 843-848, 2008.

[9] Q. Dupont and B. Maurer, "Ledgers and Law in the Blockchain," Kings Review, 2015.

[10] J. J. Sikorski, J. Haughton, and M. Kraft, "Blockchain technology in the chemical industry: Machine-to-machine electricity market," Applied Energy, vol. 195, pp. 234-246, 2017.

[11] M. Verbeek, Guide to modern econometrics: John Wiley & Sons, 2008.

[12] R. J. Rummel, Applied factor analysis: Northwestern University Press, 1988.

[13] A. C. Rencher, Methods of multivariate analysis vol. 492: John Wiley & Sons, 2003.

[14] O. Boutkhoum, M. Hanine, T. Agouti, and A. Tikniouine, "A decision-making approach based on fuzzy AHP-TOPSIS for selecting appropriate cloud solution to manage big data projects," International Journal of System Assurance Engineering and Management, pp. 1-17, 2017.

[15] H. M. Taylor and S. Karlin, An introduction to stochastic modeling: Academic press, 2014.