International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys.13, No.1 (May-23)

http://dx.doi.org/10.12785/ijcds/130192

Towards The Process Mining Applicability in The ChickenHunt Blockchain Game

Zineb Lamghari[1]

[1]*Laboratory of Innovative Technologies, High School of Technology (EST), Sidi Mohamed Ben Abdellah University, Fez, Morocco*

Received 23 Feb. 2022, Revised 31 Mar. 2023, Accepted 5 Apr. 2023, Published 30 May. 2023

Abstract: In a distributed peer-to-peer network, blockchain technology enables the secure transmission of digital assets. This is done based on intellectual agreement capabilities. Indeed, blockchain world has developed into a tool for managing transversal processes on an unbiased platform. Process mining has evolved as a well-known toolkit for comprehending different organization processes. Recently, researchers developed strategies for resolving the issue of collecting reliable data gathered from blockchains in order to improve the examination of blockchain applications employing process mining. There is yet to be a clear assessment of the utility of process mining on public blockchain event logs.
In this paper, we will validate the applicability of process mining on public blockchain event logs by treating ChickenHunt data set. ChickenHunt is a competitive game that is operated as a Decentralized application on Ethereum blockchain network.

Keywords: Public Ethereum Blockchain; Conformance Checking; Performance Analysis; Process Mining Techniques; Process Discovery; ChickenHunt Game

1. INTRODUCTION

A blockchain may be defined as a scalable, timestamped transaction data storage [1]. In this sense, blockchain has developed as a solution that enables the management of cross-organizational operations into a set of blocks using neutral framework [2], [3]. This is done based on intellectual agreement capabilities that enable the establishment and executing the defined user algorithms. Thus, blockchain is a distributed, peer-to-peer network that uses encryption to properly serve applications and manage data without the need for a trusted administrator in a veritable manner.

Process Mining has evolved as a standard toolkit for comprehending different organization processes in practice [4]. The majority of process mining efforts were devoted for process discovery [5]. However, it has been shown that process mining yields more than simply process discovery. Over time, efforts were split between conformity verification [6], performance analysis [7], process optimization [8], prediction [9], and suggestion by incorporating other disciplines like machine learning [10] and Robotic Process Automation [11]. Beyond good enough condition and procurement, process mining has expanded to cover business, transportation, healthcare, energy production, customs, insurance, mobility, user-interface design, smart devices, and airports [12]. In this context, process mining is employed in over 160 organizations, including municipalities, high-tech manufacturing, healthcare, and others.

From one hand, there are few publications and shared best practices for blockchain-based process mining applications such as [13], [14]. There aren't many people that are well-versed in the sector. Furthermore, certain process mining capabilities are currently being implemented as proposal validity, and the use cases offered by organizations as well as service providers are projects where the implementation was successfully completed. As a result, the study is constrained to relying on success stories. Thus, Process mining on public blockchain event can be appeared as a difficult challenge. There is yet to be a clear assessment of the utility of using process mining techniques in the blockchain problematic.

On the other hand, researchers developed strategies for the difficult task of collecting reliable information from blockchains in order to improve the examination of blockchain applications employing process mining. In this sense, Concepts for using extracted blockchain data were introduced, such as controlling business processes on a blockchain [2], verifying consensus protocols on Hyperledger Fabric [15], controlling blockchain apps on Ethereum [16], reviewing data recorded on the Ethereum network without targeting particular blockchain technologies of Decentralised Applications (DApps) [17], and employing process mining on a blockchain environment [18].

In this paper, we will validate the applicability of process mining on blockchain data by treating ChickenHunt data set. ChickenHunt is a competitive game that is operated as a Decentralized application on Ethereum blockchain network.

E-mail address: lamghari_zineb@yahoo.com

Because ChickenHunt intellectual agreement operate on the public Ethereum blockchain, transactions are timestamped, visible, and accessible. We make use the Ethereum Logging Framework (ELF) [19], [20] to treat ChickenHunt transactions. We obtained 715 traces and more than 138,889 events for two years and 6 months.

In more detail, we will be able to generate valuable business insights, as well as provide a clear view of the ChickenHunt's core mechanisms and regular checks behaviors and problems in (deterministic) code; data integrity is a challenging issue from the standpoint of Business Process Management (BPM) [8], and computer engineering in broad [21]. The value of this study was focused on the discussion section.

The remainder of the paper is organized as follows: Section 2 introduce the object of our case study. Section 3 illustrates our method phases that aims at mining data of the ChickenHunt game, extracted from the public Ethereum blockchain. Section 4 details our case study into the four main phases of our proposed method. In the first stage, we prepare data. Then, we filter and capture data to be adequately for the process discovery phase. Also, we apply conformance checking techniques to deduce more insights about the treated data. Last, we proceed to the performance analysis stage to present improvement propositions. Also, this section describes the findings. The section 5 summarizes the study and outlines the upcoming studies.

2. The case of ChickenHunt

ChickenHunt [1] a character-growing Integrated Development and Learning Environment (IDLE) and blockchain game, where participants receive shares and profit based on their level of participation in the game. Through reward incentives, it fosters user engagement and volunteer promotion. The objective is to populate the Ether crypto money. Furthermore, ChickenHunt is a DApp on Ethereum that is an incremental game. The goal of the game is to attract chickens by hunting and carrying out attacks on other players. Players can also improve their avatars' attack ("Upgrade Hunter"), defense ("Upgrade Depot"), and collection abilities ("Upgrade Pet"). The player is responsible for the gas costs associated with Ethereum transactions. Indeed, there are two sorts of player in the gameplay mechanic. Players can become investors of the game through certain transactions, and they can also earn money by sacrificing collected chickens for ether.

Currently, there are three versions of the ChickenHunt game. The first version 3.2 launched the fourth May 2015. The second version 3.8launched the 26th of July 2016. The last version 1.0.4 launched the 28th of January 2022. In this paper, we will treat the second version event logs. We ignored the first version because it is updated in 2016 and we ignore the latest version because of the reduced amount of available data. The data can produce partial traces (behaviors), and this is not profitable in terms of business process improvement.

[1] https://chickenhunt.io/

For several reasons, ChickenHunt was considered for this case study. Data accessibility. ChickenHunt v2 was one of the most widespread Ethereum DApps, giving in a significant volume of data to analyze. Design of an application: chickenHunt is designed so that events monitored and archived in considerable detail by a central logging contract, allowing insights into user behavior and simplifying data retrieval with ELF (in opposition to other DApps, where logging is spread over numerous contracts). Information about a subsidiary: there is information on ChickenHunt available, such as at "ingo-weber.github.io," which can be used as a starting point, for example, for conformance checking. As a result, ChickenHunt appeared to be an intriguing candidate for further investigation.

In this game, Players can be active in six roles: hunter, defender, seller, buyer, intermediate, and receiver. A hunter is a player status in which the player chases another player in order to steal a chicken. This time, the system computes the number of stolen chickens. The hunted chicken is then brought to the altar with two indications: the time it was brought to the altar and the number of chickens. When a player must protect himself against an onslaught by another player, he or she is referred to as a defender. The system calculates the number of chickens lost to the attacker at this point. A seller is a player who has the ability to sell an item or a store to another player. The system mentioned the amount paid for the item or the store in the case of the item. A buyer is a player who buys an item that improves their abilities, or a player who purchases a store from another player. The player can also enhance the abilities of his or her hunter, pet, and depot. A player who acts as an intermediary allows another player to squander his or her shares. It is critical to identify both the nominated spender and the shares that have been approved for spending. Last but not least, a receiver is a player who earns dividends from the game or another player. In addition, he can be granted permission to spend shares from another account. Furthermore, the player can earn Ethereum based on the number of chickens brought to the altar.

3. PREPARE YOUR PAPER BEFORE STYLING

In this section, we present our method phases for mining ChickenHunt data stored on the public Ethereum blockchain (see figure 1). Our method consists of the following four phases: 1- Data preparation, 2- Process Discovery, 3- Conformance checking 4- Performance analysis.

A. Data preparation

To extract the data in this paper, it is necessary to use the publicly available Ethereum Logging Framework [1] (ELF). Analysts use ELF to extract, convert, and format data from Ethereum network blocks, transactions, log entries, and intellectual agreement. ELF accepts a manifest as input. This includes instructions on what data to collect and how to handle it. In addition, we must define a manifest file for the treated event logs based on the data source that provided us with all the log entry definitions. The manifest's execution

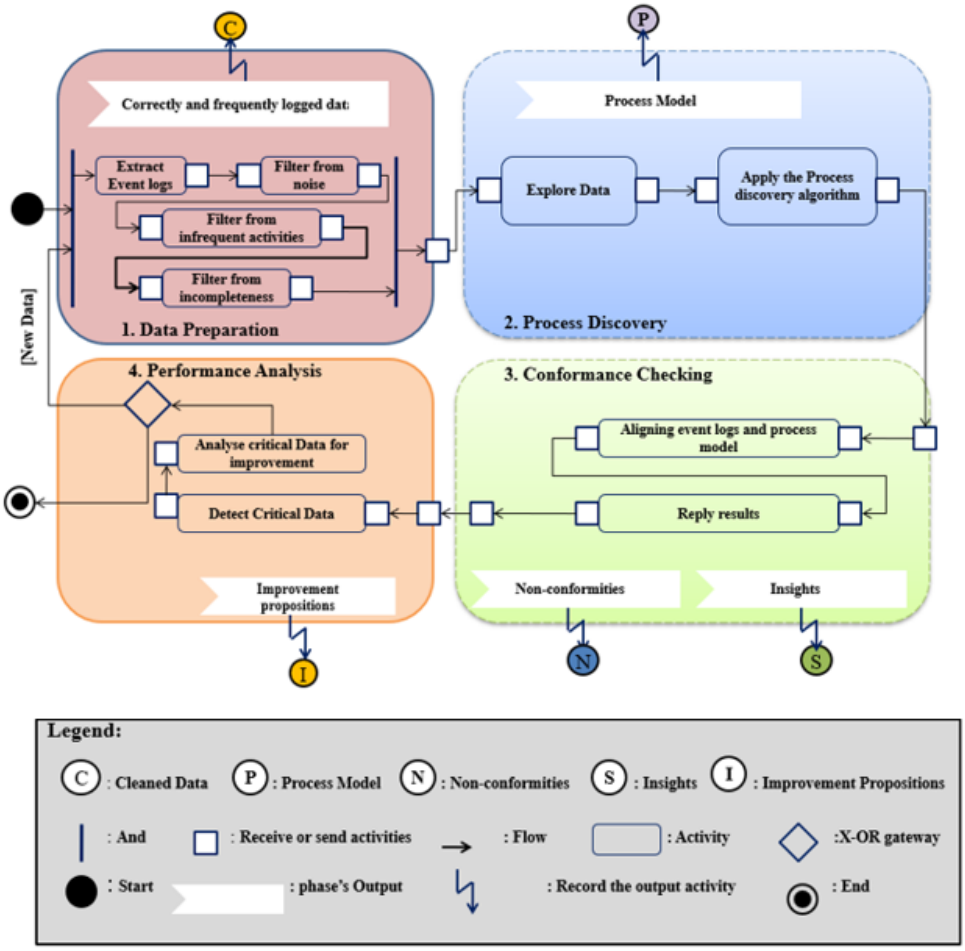

Figure 1. Our method's phases overview

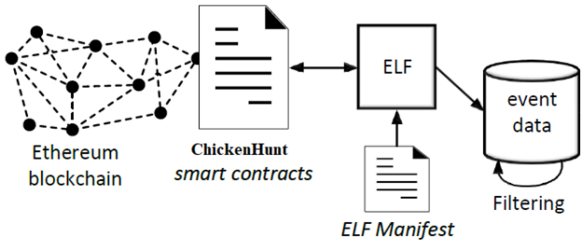

Figure 2. Data preparation

generated an event log in the XES format [22], with a XES event encompasses the information from ChickenHunt's log entry for each log. The events can then be classified as traces related to the treated environment (see figures 2 and 3).

B. *Process discovery*

Process discovery is a technique of process mining that starts with an event record and ends with a meaningful process model (see figure 4). This is done using historical event data to discover what is happening in the

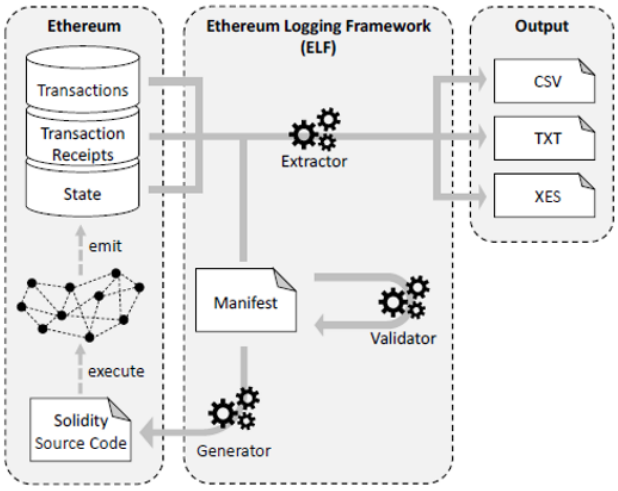

Figure 3. The components of the Ethereum Logging Framework (ELF) [19]

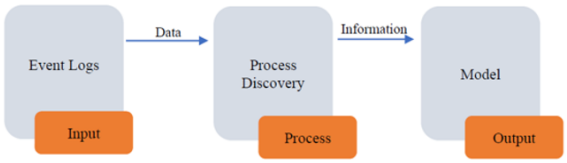

Figure 4. Process discovery technique

actual process [23]. To enhance resource management, the actual process may determine non-conformities, minimize non-value-added operations, decrease waste, and visualize wasteful waiting and rework. The model's finding serves as an initial stage for analysis; it provides a solution to the query, "What happened?" When the events are tied to the revealed model, other studies such as checking conformity, performance analysis, finding bottlenecks, removing non-value-added operations, decreasing waste, resource allocation, forecasts, and recommendations may be done [24].

The behavior observed in the given event log should be represented by the model. To operationalize the representation, four quality criteria must be met [24]: Fitness is related to the model's ability to replay all event logs. Precision is linked to underfitting; a faulty precise model allows behavior that does not appear in the event log. Overfitting is related to generalization; process models are expected to generalize the behavior observed in event logs. The principle of simplicity is referred to as "Occam's Razor". It is correlated to the process model that is being used for the simplest explanation of the underlying process.

C. Conformance checking

The examination of the relationship between the intended behavior of a process as explained in a process model and event logs recorded during the process's execution is defined as conformance checking (see figure 5). Figure 10 depicts event logs and models (made by humans or discovered from event logs) as inputs and diagnostics as outputs (where the supplied model and logs do not match). The purpose is to detect similarities and differences between the prepared and mined behaviors [24]. In this context, model-model conformance checking implies that the event logs correspond to reality, whereas log-model conformance checking doesn't seem to [25].
Two points of view should be supported by conformance checking methods. The first point to examine is that the current model may not adequately reflect reality and must be rectified or enhanced. The second step is to acknowledge that some circumstances diverge from the model and that a stronger control mechanism is necessary to compel better compliance. Systems, for example, can be configured to prevent unexpected behavior. Indeed, conformance checking necessitates the use of a normative process model. We supplemented it with insights gleaned from discovery and conformance checking in cases where the information in the repository, which comprises event logs of DApps deployed on public blockchain networks, was insufficiently

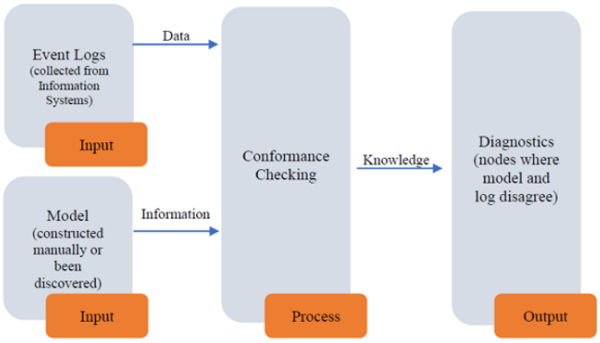

Figure 5. Conformance checking

detailed or precise. Figure 6 depicts the resulting process model. Section 4 discusses additional information on initial discrepancies.

D. Performance analysis

A process's or organization's performance can be defined in a variety of ways. Typically, three performance parameters are identified: time, money, and quality. Various crucial performance indicators can be established for each of these performance characteristics. When the time dimension is considered, the following performance indicators can be identified: The whole time from the formation of the case to the completion of the case is referred to as the lead time (also known as flow time). Service time is the amount of time spent working on a case. The waiting time is the amount of time it takes for a resource to become accessible in a case. The synchronization time is the period during which an activity is not completely activated and is waiting in the case of an external trigger or another parallel branch. Cost-related performance indicators can also be defined. The time measure will be the subject of our case study.

4. Exploring the ChickenHunt Game

In this section, we will go over the data extraction and pre-processing techniques. The focus of the study is data analysis, which includes data exploration, process discovery, conformity checking, and performance analysis. We gathered information related to 715 players fading from the period of 2017-12-25 to 2021-05-11. We carried out the retrieval of data from 2018-06-25 to 2021-02-16 (see Tables I and II).
All the data and code used in this work are freely available to the public. for replication purposes, including the source code of ELF [2], the manifest, the normative process model and the accompanying XES log [3], the source code of chickenHunt [4], and the data on the public Ethereum blockchain. We uploaded the event log into multiple process mining tools to examine the players' behavior, but we'll focus on the ProM results here. All the experiments were performed

[2]https://ingo-weber.github.io/dapp-data/elf-scripts/ChickenHunt.ethql
[3]https://bit.ly/3z2l2Fw
[4]https://bit.ly/3Dii0PY

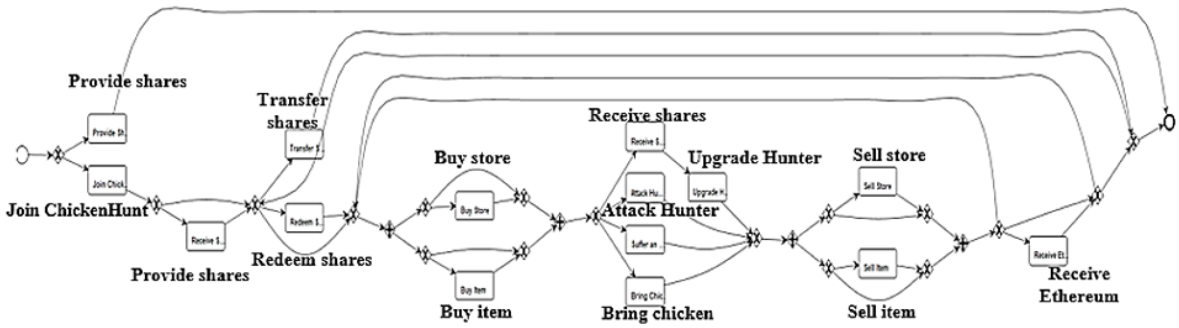

Figure 6. ChickenHunt normative model

on a laptop with the specifications : PC Intel(R) Core(TM) i7-8550U CPU 1.80GHz.

Each trace in the ChickenHunt log documents occurrences throughout a ChickenHunt player's game, i.e., the actions that a ChickenHunt player does or participates in while playing the game. The following features are shared by all traces:

- The concept: name represents the ChickenHunt player's address. Furthermore, each event has a set of typical characteristics;
- concept: the event's class is denoted by the name;
- timestamp: the timestamp of the block containing the event;
- lifecycle: transition is each event's lifecycle transition, which is set to completed by default for all events. This property is present to guarantee interoperability;
- blockNumber is the number of the block that containing the event;
- transactionIndex is the index of the transaction that contained the event; and
- logIndex is the index of the log that contained the event.

During a player's game, the following events can occur:

- Become a Chicken member;
- Receive Ethereum from Altar - The gamer earns Ethereum depend on the number of chickens that he, or she brought to the altar;
- Bring Chicken to Altar - The gamer brings a hunted chicken to the altar;
- Attack Hunter is the gamer attacks another gamer to steal chicken;
- Suffer an Attack - The gamer must protect himself against an onslaught by another player;
- Upgrade Hunter is the gamer enhances the aptitude of his or her hunter;
- Upgrade Hunter - The gamer enhances the aptitude of his or her hunter;
- Upgrade Depot - The gamer improves his or her depot;

TABLE I. Traces summary

Traces	487
Events	94,452
Event Classes	3
Attributes	10
Variants	296
Events per Trace	197,598
First Event	2018-06-26
Last Event	2020-12-262

- Upgrade Pet - The gamer improves the aptitude of one of his or her pets;
- Buy Store - The gamer purchases a store from another player;
- Sell Store - The gamer sells a store to another player;
- Redeem Shares - The gamer (or the game) cashes out shares;
- Transfer Shares - The gamer (or the game) transfers shares to another player;
- Receive Shares - The gamer receives shares from the game or another player; and
- Receive Shares Approval - The gamer is given permission to spend shares from another account.

The most typical player behavior is demonstrated: 107 players out of 715 cases where the player joins the ChickenHunt and never engage in anything else. Several frequent traces indicate players joining and getting attacked (one or more times) with no further actions. Some players follow a similar strategy, except they start by bringing hens to the altar. Individual traces of players who are more engaged in the game are significantly varied. There are 402 separate traces for each of the 715 cases (see Figure 7).

Figure 7 shows that 313 traces are frequently executed. On the other hand, the most common behavior, with just minor variations "Join ChickenHunt" and with a score of 107 traces and 14,97% of occurrences. Figure 8 is not meant to be understood, but it does convey an idea of the trace variability, where 402 of the 715 variants They are one-of-a-kind, encompassing 66.99 percent of all behaviors.

TABLE II. ChickenHunt Data overview

Events Name	Events attributes	Common attribute
Events	concept:name	concept:name
Event Classes	time:timestamp	
Attributes	lifecycle:transition	
Variants	Completed	
Events per Trace	blockNumber	
First Event	transactionIndex	
Last Event	logIndex	
Upgrade Hunter		
Upgrade depot		
Upgrade pet		
Buy Store		
Sell Store		
Redeem Shares		
Transfer Shares		
Receive Shares		
Receive Shares Approval		
Provide Shares Approval		

We examined the sequence and frequency of the various sorts of improvements in Figure 8's immediately following graph. Enhancing the hunter appearance is definitely the most widely used option, and it is also the most common first and last improvement. In other words, while active gamers can enhance their pets and depots, they usually return to improving their hunter. These findings can be valuable to each game creator.

The dotted chart illustrates all related event logs. In the analysis step, we concentrate on the events of joining, assaulting, and being attacked (see Figure 9). It can be seen that just a few people attack others, yet a significant number of players get attacked. Furthermore, the attacks appear to occur in coordinated waves, as shown by the vertical arrangement in the dotted chart. The causes of such waves might be related to the gas costs (and hence the charges) per transaction on Ethereum (see, for example, etherscan); a visual comparison of the timelines shows that increased gas costs on Ethereum may well coincide with times without ChickenHunt assaults. The attackers presumably kidnapped chickens from regular users, carried them to the altar, and got Ether in exchange, all of which included transactions with related charges. If the Ether returns are not sufficiently large, the charges can result financial loss for this activity.

The longest waiting time during the transition between the following tasks was the most noticeable outcome of conformance checking (see figures 10 and 11):

- From attack hunter to Receive Ethereum from altar.
- From Bring chicken to Altar Receive Ethereum from altar.
- From attack shares to Redeem Shares.
- From Buy Item to Redeem Shares.
- From Buy store to Redeem Shares.
- From receive Ethereum from Alter to Redeem Shares.
- From Sell Item to Redeem Shares.
- From Sell store to Redeem Shares.
- From Suffer an attack to Redeem shares.
- from Redeem shares to sell item.
- From attack hunter to suffer an attack.
- From receive shares to suffer an attack.
- From attack hunter to transfer shares.
- From Bring chicken to transfer shares.
- From Buy Item to transfer shares.
- From Buy Store to transfer shares.
- From Join chickenHunt to transfer shares.
- From receive shares to transfer shares.
- From Sell Item to transfer shares.
- From Sell Store to transfer shares.
- From Suffer an attack to transfer shares.
- From Upgrade hunter to transfer shares.

5. Discussion

In this work, the approach for mining ChickenHunt event logs extracted from the Ethereum blockchain that hands over four main results. These results provide a decision-making support for the developer and enable to improve the chickenHunt game that respects the player's requirements and the internal or external regulations.

The first result represents the correct and the frequent event. This phase is similar to the studies published in [23], [25], [26]. Indeed, we applied several filtering mechanisms in order to extract data. In this approach, we use four filtering functions, to clean event data from noise, incompleteness, chaotic and infrequent behavior.

Based on the data preparation phase, the ChickenHunt lists 17 behaviors, only 15 of which could be observed following data retrieval. This mismatch can be explained in part by

Figure 7. Some of the most frequent variants

events such as the fork event not really being affected during the application's lifespan. In the event that an assault cannot be matched across several rounds, the option to fork acts as a last resort and serves as a final resolution process.

The second result is about discovering chickenHunt process model (see Figure 6). The novelty, here, is to introduce the process discovery on processes extracted form blockchain.

The third result is about comparing the normative and discovered models. Here, we observed that 389 cases are perfectly fitting, while 326 cases non fitting. On the other hand, 709 are properly started. The period of observation is 31,80 months. The most eye-catching outcome of conformance testing was the longest waiting time as discussed in section 4. Indeed, the non-conforming repeated create problematic events during replaying event logs on the discovered model.

The fourth result gives performance analysis for the chickenHunt event logs. We had no intention of using process mining to test for system vulnerabilities during the engineering or pre-deployment phases. Here, we demonstrated that process mining would be used to detect flaws and performance concerns in blockchain apps after they have been deployed (depended on real player actions), allowing programmers to remedy holes or codify anomalous updated behaviors. However, the used methods for checking vulnerabilities at design time are still vital, especially for DApps, but they may be supplemented by analysis like ours.

In reality, there is one branch of study which addresses a comparable problematic, and it is user activity analysis. Human computer interaction is documented in this arena to evaluate the user in relation to a certain study goal. The method is utilized in a variety of fields, including e-

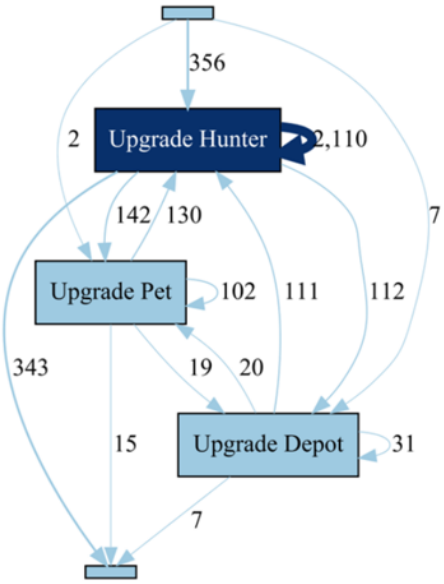

Figure 8. Frequency of the different types of upgrades

commerce and online social network research, to provide services such as recommendation systems [27] and to evaluate social behavior [28]. There is no example of user activity analysis on blockchain event logs using process mining techniques to validate the applicability of process mining on blockchain events.

6. Conclusion

In this article, we did a case study on process mining on the blockchain application ChickenHunt. For that purpose, we utilized ELF [19] to extract data from ChickenHunt version 2 during. We explored the data using process mining methodologies and tools, discovered models for a collection of variations, and performed compliance verification and effectiveness.

Last, we suggested a series of ideas for expanding the chickenHunt game. This data might help developers understand why gamers quit early and be utilized to make changes.

In conclusion, we argue that there are scientific proofs for the use of process mining on blockchain data. Future studies may be conducted to evaluate alternative apps that may operate on other blockchains, or on the same blockchain, such as Forsage and cryptokitties.

Acknowledgement

This work is supported by the National Center for Scientific and Technical Research (CNRST) in Rabat, Morocco.

References

[1] W. Yang, C. Peng, W. Bing, W. Chengpeng, and Z. Yang, "A trustable architecture over blockchain to facilitate maritime administration for mass systems," *Reliability Engineering and System Safety*, vol. 219, pp. 108 246–108 246, 2022.

[2] R. Hamzah, B. Ary, A. Zaldy, and A. Adhi, "Permissioned blockchain for business process visibility: A case of expenditure cycle," *Procedia Computer Science*, vol. 197, pp. 336–343, 2022.

[3] V. Wattana, B. Zhuming, and H. Danupol, "Blockchain technologies for interoperation of business processes in smart supply chains," *Journal of Industrial Information Integration*, vol. 26, 2022.

[4] l. C. Silva and d. S. M. Mira, "Research contributions and challenges in dlt-based cryptocurrency regulation: a systematic mapping study," *Journal of Banking and Financial Technology*, vol. 6, pp. 63–82, 2022.

[5] W. Philipp, P. Lukas, R. Kate, and M. Jan, "Causal process mining from relational databases with domain knowledge," *arXivLabs*, 2022.

[6] J. Munoz-Gama, "Conformance checking and diagnosis in process mining," *arXivLabs*, 2016.

[7] F. Rokhman and A. Rachmadita, "Business process analysis of programmer job role in software development using process mining," *Procedia Computer Science*, vol. 197, pp. 701–708, 2022.

[8] C. Timea, K. Alex, R. Tamas, and A. Janos, "Data-driven business process management-based development of industry 4.0 solutions," *CIRP Journal of Manufacturing Science and Technology*, vol. 36, pp. 117–132, 2022.

[9] P. Patil and R. Hiremath, "Big data mining—analysis and prediction of data, based on student performance," in *Pervasive Computing and Social Networking*, G. Ranganathan, R. Bestak, R. Palanisamy, and Á. Rocha, Eds. Singapore: Springer Singapore, 2022, pp. 201–215.

[10] R. Engel, P. Fernandez, A. Ruiz-Cortes, and A. Megahed, "Conformance checking and diagnosis in process mining," *Information Systems and e-Business Management*, vol. 20, pp. 199–221, 2022.

[11] H. Adrian, P. Tobias, S. Franz, W. Jonas, F. Marcus, and W. Axel, *4 Process selection for RPA projects: A holistic approach*. Berlin, Boston: De Gruyter Oldenbourg, 2021, pp. 77–90.

[12] V. F. Agnes, V. Istvan, and K. Istvan, "Multi-level process mining methodology for exploring disease-specific care processes," *Journal of Biomedical Informatics*, vol. 125, 2022.

[13] D. Jia, Z. Liu, Z. Zhang, and J. Ye, "Research on security vulnerability detection of smart contract," in *2021 International Conference on Big Data Analytics for Cyber-Physical System in Smart City*, M. Atiquzzaman, N. Yen, and Z. Xu, Eds. Singapore: Springer Singapore, 2022, pp. 957–963.

[14] K. Philipp and B. Freimut, "Blockchain-based cross-organizational execution framework for dynamic integration of process collaborations," in *Wirtschaftsinformatik*, 2020.

[15] D. Megias, Q. Nasir, I. A. Qasse, M. Abu Talib, A. B. Nassif, and J. Ojeda-Perez, "Blockchain technologies for interoperation of business processes in smart supply chains," *Journal of Industrial Information Integration*, vol. 2018, 2018.

[16] K. Desen, C. Benoit, R. Valentin, S.-B. Marcelo, N. Sonam, A. Merlinda, A. Ioannis, N.-P. Matias, F. David, and K. Aristides, "Smart contracts in energy systems: A systematic review of fundamental approaches and implementations," *Renewable and Sustainable Energy Reviews*, vol. 158, p. 112013, 2022.

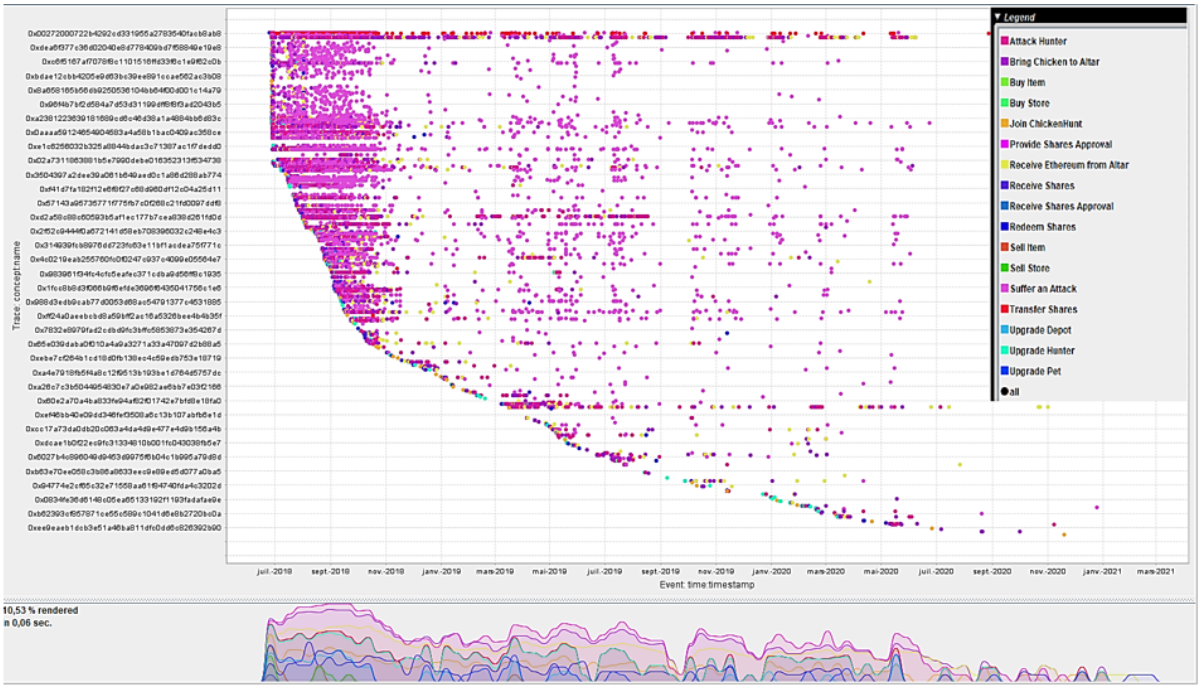

Figure 9. Dotted chart showing all 138,889 events

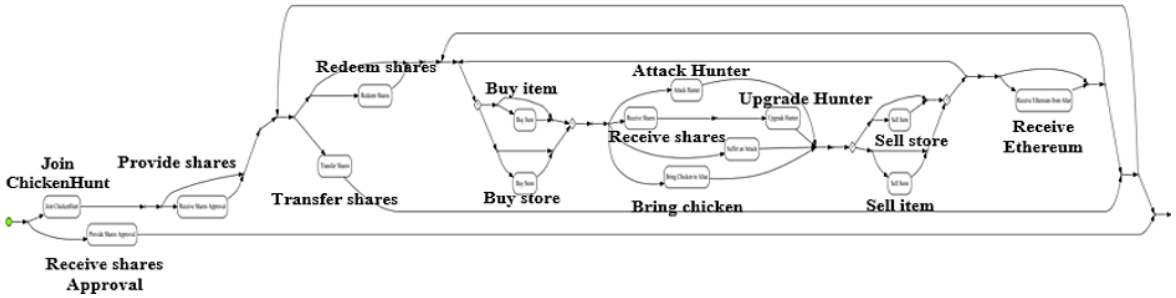

Figure 10. Matching event logs and process model

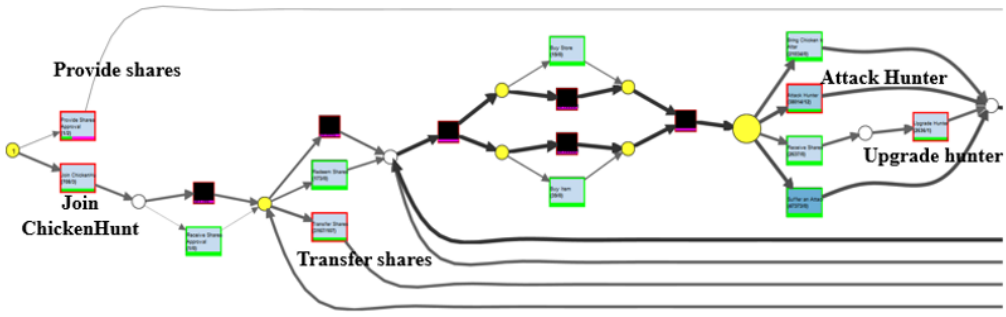

Figure 11. Process tree representation using standard configuration

[17] A. S. Nicole, C. L. Jason, C. L. Hans, R. Maria, E. Justina, P. Christian, V. Dennis, S. Quirino, G. Maria Cristina, and A. Paulyn Jean, "Clinical interactions in electronic medical records towards the development of a token-economy model," *Procedia Computer Science*, vol. 196, pp. 461–468, 2022.

[18] M. Müller, A. Simonet-Boulogne, S. Sengupta, and O. Beige, "Process mining in trusted execution environments: Towards hardware guarantees for trust-aware inter-organizational process analysis," in *Process Mining Workshops*, J. Munoz-Gama and X. Lu, Eds. Springer International Publishing, 2022, pp. 369–381.

[19] R. Hobeck, C. Klinkmüller, H. M. N. D. Bandara, I. Weber, and W. M. P. van der Aalst, "Process mining on blockchain data: A case study of augur," in *Business Process Management*, A. Polyvyanyy, M. T. Wynn, A. Van Looy, and M. Reichert, Eds. Springer International Publishing, 2021, pp. 306–323.

[20] K. Christopher, W. Ingo, P. Alexander, T. An Binh, and A. Wil van der, "Causal process mining from relational databases with domain knowledge," *arXivLabs*, 2020.

[21] I. Weber and M. Staples, "Programmable money: next-generation blockchain-based conditional payments," *Digital Finance*, vol. 4, pp. 109–125, 2022.

[22] G. Acampora, A. Vitiello, B. Di Stefano, W. van der Aalst, C. Gunther, and E. Verbeek, "Ieee 1849: The xes standard: The second ieee standard sponsored by ieee computational intelligence society [society briefs]," *IEEE Computational Intelligence Magazine*, vol. 12, no. 2, pp. 4–8, 2017.

[23] M. Niels, V. H. Greg, and J. Gert, "Daqapo: Supporting flexible and fine-grained event log quality assessment," *Expert Systems with Applications*, vol. 191, 2022.

[24] W. van der Aalst, *Process mining: Data science in action*. Springer-Verlag Berlin Heidelberg, 2016.

[25] S. J. J. Leemans, D. Fahland, and W. M. P. van der Aalst, "Daqapo: Supporting flexible and fine-grained event log quality assessment," *Expert Systems with Applications*, vol. 17, pp. 599–631, 2018.

[26] Z. Lamghari, R. Saidi, R. Maryam, and M. D. Rahmani, "Chaotic activities recognizing during the pre-processing event data phase," *International Journal of Business Intelligence and Data Mining*, vol. 20, pp. 412–439, 2022.

[27] H. AlQaheri and M. Panda, "An education process mining framework: Unveiling meaningful information for understanding students learning behavior and improving teaching quality," *Information*, vol. 13, 2022.

[28] L. Jr. Jackie, L. Siyuan, and S. Heshan, "Seems legit: An investigation of the assessing and sharing of unverifiable messages on online social networks," *Information Systems Research*, vol. 33, pp. 978–1001, 2022.

Zineb Lamghari Assistant Professor at the Higher School of Technology (EST), Sidi Mohamed Ben Abdellah University, Fez, Morocco. She is a member of the Innovative technologies Laboratory (LTI) of EST. She received her PhD in Software Engineering from the Mohammed V University in Rabat, Morocco in 2021. Her research interests are mainly focused on information systems, software development, business process, process mining, business intelligence. She has published several papers in international journals, conferences, and workshops.