Modeling Hackathons Platform for a Token Economy

Fabio dos Santos Cardoso
School of Business and Economics / Data Science Knowledge Center, Nova University Lisbon
Portugal
fabio.cardoso@novasbe.pt

Joao Gama Oliveira
School of Business and Economics / Data Science Knowledge Center, Nova University Lisbon
Portugal
joao.oliveira@novasbe.pt

Helder Vasconcelos
TAIKAI
Portugal
helder@layerx.xyz

Leid Zejnilovic
School of Business and Economics / Data Science Knowledge Center, Nova University Lisbon
Portugal
leid.zejnilovic@novasbe.pt

Abstract

Hackathons have become a popular tool to connect companies seeking innovations with talented challenge-seeking problem-solvers. An important factor for the success of hackathons is the effective system for distributing rewards and value and incentivizing participation. In pursuit of an efficient reward system, some digital platforms for facilitating hackathons are resorting to programmable blockchains and token-based rewards. Integration of tokens may stimulate participation, increase trust, increase the expected value of the rewards over time, and ease the distribution of rewards. However, designing such systems is hard, and anticipating the system-level outcomes is even more challenging. In this work, we develop an agent-based model to study potential scenarios in facilitating interactions between creators, companies, and the platform. Such simulations allow digital platforms to design and test their incentive layers and optimize their reward distribution with minimal cost.

CCS Concepts

• **Blockchain, Agent-based model, hackathons, token economy**;

Keywords

modeling, digital platform, tokenization

ACM Reference Format:
Fabio dos Santos Cardoso, Joao Gama Oliveira, Helder Vasconcelos, and Leid Zejnilovic. 2024. Modeling Hackathons Platform for a Token Economy. In *2024 6th Blockchain and Internet of Things Conference (BIOTC 2024), July 19–21, 2024, Fukuoka, Japan.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3688225.3688233

1 Introduction

Hackathons are intensive events where participants collaborate in small teams to ideate, develop, and present solutions to specific problems within a limited time frame [1], [2]. These events have gained popularity for fostering innovation, creativity, and rapid prototyping in various domains. With the advancement of technology, digital platforms have emerged as facilitators of hackathons, providing online spaces for participants to engage in collaborative problem-solving activities [3], [4]. These platforms enable remote participation, asynchronous communication, and the organization of virtual hackathons that bring together individuals from diverse backgrounds to work on innovative projects [5]. However, digital platforms for hackathons also face challenges, including ensuring effective collaboration, maintaining participant engagement, and managing the complexity of virtual interactions [5], [6]. Issues such as socio-technical constraints, asynchronous communication barriers, incentives for participation, and the need for robust data collection frameworks pose challenges to the seamless operation and continuity of these platforms.

Blockchain and tokenomics are emerging as potential ways to address these challenges and enhance the effectiveness of digital platforms for hackathons. The inclusion of blockchain in digital platforms for hackathons opens up opportunities for exploring new ways of data management, enhancing security, and promoting transparency [7-12]. Blockchain technology helps ensure the integrity of project submissions, provides transparent voting mechanisms, and enables secure transactions within the hackathon ecosystem. Additionally, the integration of tokenomics principles can help incentivize active participation, reward contributions, and foster inclusive community engagement among hackathon participants [13 - 16].

The literature on the impact of tokenization on the performance of hackathon platforms is scarce. Yet, the hackathon hosting platforms must develop strategies for tokenization and experiment with them to find the optimal one that incentivizes the desired behaviors and outcomes. In this study, we propose an agent-based model for exploring strategies for introducing tokens to digital platforms for hackathons. We demonstrate in a simplified example how these models function and what the emerging results are. Such an approach allows platform owners to design and simulate randomized tests and then launch targeted campaigns and compare simulated data with real-world outcomes. This approach renders the proposed agent-based models as invaluable tools in searching for optimal token launching strategies, increasing the value and the use of blockchain but also enhancing the functionality and effectiveness

BIOTC 2024, July 19–21, 2024, Fukuoka, Japan

ACM ISBN 979-8-4007-1700-0/24/07
https://doi.org/10.1145/3688225.3688233

of digital platforms for hackathons, ultimately fostering innovation, creativity, and impactful problem-solving.

In the next subsection, we provide a brief comparison of potential methodologies for modeling the interactions in complex systems, such as one of digital platforms for hackathons. We then introduce our chosen methodology, followed by results and a short discussion with conclusions.

2 Modeling approaches for the token economy: types and features

To develop models for strategic choices in the token economy, we consider three types of modeling approaches: (I) system dynamics, (ii) network, and (iii) agent-based modeling. In this subsection, we compare these approaches in terms of their strengths, applications, and limitations. To illustrate these three approaches, consider traffic flow as a system. System dynamics models can be used to capture the overall flow of vehicles, represented by stocks (number of vehicles) and flows (entering and exiting highways). Network models can be used to represent roads as links and analyze how congestion spreads. Agent-based models can be used to simulate the decision-making processes of individual drivers, allowing traffic patterns to emerge. The selection of an appropriate type of modeling approach is critical for allowing better understanding and studying the function of complex systems in the real world.

2.1 System Dynamics Model

System Dynamics (SD) is a modeling paradigm that employs sets of differential equations to represent the non-linear behavior of complex systems in terms of flows, feedback loops, and stocks or state variables [17], [18]. SD modeling is particularly useful for modeling population flows, financial balances, and other systems with relatively simple interactions. However, it has limited ability to represent complex agent and system interactions. The Lotka-Volterra model provides an illustrative example of system dynamics modeling paradigms. It describes a dynamical system consisting of two interacting species, one of which is a predator and the other a prey [19]. Compartmental models, which refer to various systems, including ecological, epidemiological, and sociological systems, are also worth mentioning [20]. These models are characterized by populations of individuals belonging to specific categories, such as chemical species, opinions, or epidemic states.

2.2 Network Model

Network (or graph) models can be conceptualized as maps that capture the interconnections of subsystems [21]. They can be thought of as sets of nodes (representing entities such as populations) connected by edges (representing interactions such as trade or communication). These connections are not just passive lines; they can be one-way (e.g., the flow of resources) or weighted to reflect the strength of the interaction, and they go beyond static links. In contrast to traditional models, which assume a "well-mixed" environment where all elements interact with each other, network models reveal the hidden influence of the structure of the network itself [22]. By analyzing the intricate web of connections, these models can show how specific arrangements of nodes and edges can amplify or dampen effects, often leading to unexpected behavior within the system. Network models are a valuable tool for understanding complex real-world phenomena [23], including the spread of ideas on social media, the dynamics of disease outbreaks, and the flow of money in an economy.

2.3 Agent-based Model

Agent-based modeling (ABM) is a valuable approach that simulates detailed interactions of autonomous agents within a system and the emergent outcomes in the larger system [24]. These agents can represent different actors, from traders in a financial market to prey and predators in an ecosystem. In contrast to traditional mathematical models of system dynamics, ABM allows for bottom-up design, where individual agent behaviors and decision rules, sometimes including randomness, collectively generate complex system dynamics, usually by computer simulation [25]. These interactions can be competitive, cooperative, and adaptive, often resulting in unexpected behaviors that emerge from the local interactions [26]. These behaviors can provide valuable insights into real systems [27]. Furthermore, some ABM models incorporate learning algorithms, such as neural networks, that allow agents to evolve and adapt over time, further enhancing the model's realism. Additionally, the agents can reside on top of generic complex networks, where connections may represent interactions such as information sharing or trade deals. By bridging the gap between individual actions and system-wide outcomes, ABM offers valuable insights into the intricate workings of complex systems.

2.4 Summarized features

The strengths, applications, and limitations of the three models must be considered to define the better type to be applied in the token economy context. Regarding the efficiency gains resulting from interactions of autonomous agents, a model must identify potential gains unrealized on a simple view. In this way, table 1 summarizes those features. In sequence, the type most adequate for the token economy is selected.

The Agent-Based model has been selected for this study due to its capacity to provide simulations from individual agents and their interactions. This deterministic feature distinguishes it from the other two options. Additionally, the granularity capacity that allows the understanding of behaviors at the individual level is necessary to address the research about the token economy and its contexts. The proposed model and the specific context where it has been tested are presented in the next subitem.

3 An Agent-Based model of a tokenized freelancing

The proposed Agent-Based Model lays the groundwork for further exploration of tokenized freelancing platform dynamics. By simulating various scenarios and parameter settings, the model can provide valuable insights into (a) optimal matching algorithm design for project success, (b) the impact of platform fees on creator and company participation, (c) the role of reputation scores in fostering trust and project completion rates. Under this ongoing perspective, the model's features and implementations are presented below.

Table 1: Summarized features of mathematical models

Feature/Model	System Dynamics	Network	Agent-based
Strengths	Effective in capturing the emergent behavior of complex systems with feedback loops;	Efficiently represent interconnected systems using graphs (nodes and edges);	Provides a detailed, ”bottom-up” view of a system by simulating individual agents and their interactions;
	Well-suited for modeling flows, quantities, and stocks within a system;	Allow analysis of network structure and its impact on system dynamics;	Captures the emergence of complex system behaviors from individual-level actions;
	Highly efficient for simulating large-scale system behavior.	Applicable to diverse systems with well-defined interactions between entities.	Well-suited for modeling systems with autonomous decision-making entities.
Applications	Population dynamics;	Modeling disease spread through social networks;	Market simulations with diverse actors (traders, investors);
	Financial modeling;	Analyzing information flow and communication networks;	Modeling crowd behavior and social dynamics;
	Supply chain management;	Simulating economic or transportation networks.	Simulating biological systems with individual organisms.
	Policy analysis in various domains.		
Limitations	Limited ability to represent intricate agent interactions and individual decision-making processes.	May not capture the internal dynamics of individual nodes within the network.	Can be computationally expensive for very large or complex systems with many agents;
			Requires careful design of agent behaviors and decision-making rules.

3.1 Features and interactions

This work proposes a preliminary agent-based model (ABM) to explore the core interactions within a tokenized freelancing (hackathon) platform. The model focuses on three primary agent types: Creators (freelancers or hackers), Companies (clients), and the Platform (facilitator).

The agents are described according to their functions. First, the model considers the Creators as individuals who offer services with varying skill levels (S) impacting project success probability (P_success). Those agents act by applying to a project development proposed by the platform and companies according to their skill level.

Second, the agents named “Companies” propose projects and choose token rewards (R_reward) based on project complexity.

Third, the platform acts as an intermediary, matching creators with projects. As an intermediate, the Platform facilitates project development and handles financial transactions, such as fee collection. In addition, this agent (i) analyses and defines the efficiency of problem-solving, (ii) establishes reward distribution, (iii) proposes financial viability, and (iv) adjusts parameters for optimal conditions.

The platform benefits from a matching algorithm that considers the skill (S) and the reputation (R) of the creator, together with project requirements, to optimize the pairing of creators and projects. It generates revenue for its operation by deducting a fee (F_platform) from project rewards.

Figure 1 presents the general interactions between the agents that may occur on the platform, as described below:

I. Companies-Platform (circular bullets, black dashed arrow): Companies submit project proposals with R_reward to the platform.

II. Platform-Companies (squared bullets, black continuous arrow): The platform selects the project suitable for hosting on the platform.

III. Platform-Creators (“v” bullets, grey continuous arrow): The platform utilizes a matching algorithm to connect creators with suitable projects based on their skill sets and project requirements.

IV. Creators-Platform (pointed bullets, grey dashed arrow): Creators submit project completion notifications to the platform, triggering R_reward disbursement (minus F_platform).

V. The model dynamics, in this initial version, focuses on core interactions and excludes market fluctuations and agent population changes for simplicity. Future iterations can incorporate these elements for a more comprehensive analysis. For now, the model implementation shows outputs that explain the involved agents’ performances. By measuring and improving those results, it will be possible to extract further gains from the agents’ interactions in hackathons. Preliminary outcomes of the developed models are described in the next sub-subsection.

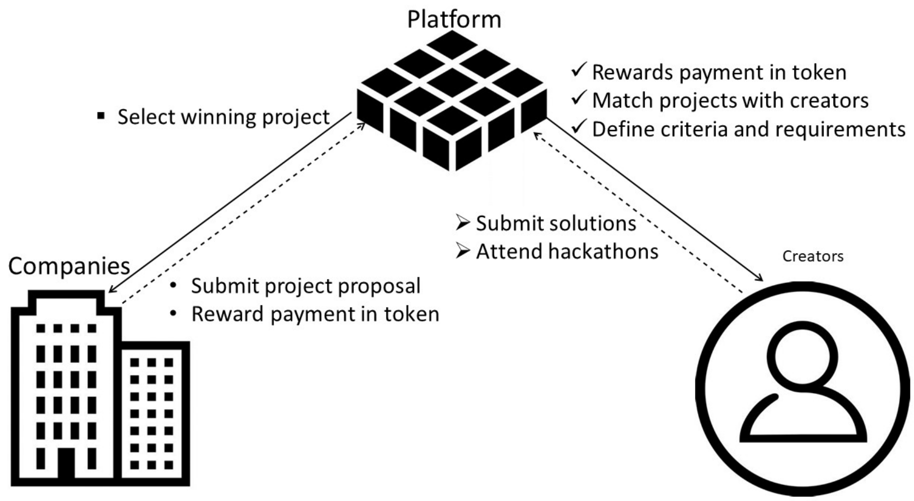


Figure 1: Schematic representation of the agent-based model, with the three agents and their interactions.

3.2 Implementation and Preliminary Results

Algorithm 1 illustrates a single simulation of the basic agent-based model that models a series of hackathons.

To illustrate an implementation of this algorithm (see Figure 2), a total of 100 creators and 10 companies were set in the simulation. Moreover, for the sake of simplicity, each company published only one project. Each project has an associated reward, given by a random uniform number between 10 and 100. Finally, each creator solves (or successfully develops) a project according to their accuracy (or skill), a random uniform number between 0 and 1. Note that this uniform distribution is made assuming that each participant in a hackathon is already endowed with problem-solving knowledge that is higher than that of an average citizen for a given problem. Therefore, the uniform distribution is chosen to introduce variability in what would likely be the right tail of the general population. While this approach introduces variability, one might also consider dynamics where hackers can improve their accuracy over time based on their past performance. For example, they could increase their accuracy by a certain amount each time they successfully solve a project. In this study, we consider the outcomes with and without this additional learning factor. But we start by considering the hacker's accuracies as simply random uniform.

The algorithm's implementation used Python version 3.10.12 (https://www.python.org/), in conjunction with the cadCAD package version 0.4.23 (https://cadcad.org/). The illustrated run was completed in a couple of seconds. This running time should be considered when considering the potential for the model to escalate and run large amounts of data.

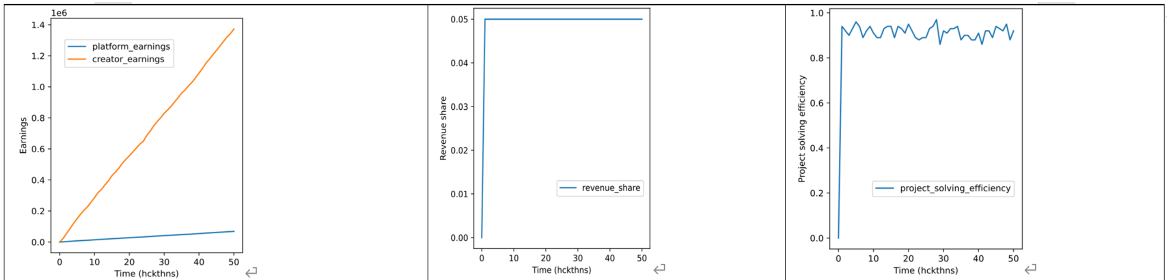
Considering the results obtained by running the algorithm of Algorithm 1 for 50 times steps (where the time unit is the realization of a hackathon), we can measure, in this basic implementation, the evolution over time of four important quantities computed: the platform earnings, the creator earnings, the platform revenue share, and the project solving efficiency. As the algorithm shows, the earnings of the platform come from a commission of 5% for each solved project. Whenever a creator solves a project, 5% of the reward assigned to that project is allocated to the platform. On the other hand, the creator's earnings come directly from solving projects, i.e., when a creator solves a project, they earn the value of the reward corresponding to that project. In what follows, we describe these measures shown in Figure 2, in the situation when the reward pool (or what can be given to a creator) is infinite.

Figure 2.a. plots the cumulative platform earnings over time as well as the cumulative creator earnings; the linear time growth of these quantities is naturally expected: since we repeat the code of Algorithm 1 up to 50 hackathons, then the evolution over these time steps of the cumulative earnings expectedly fluctuates around a linear growth coming from the average earnings in each time step.

Figure. 2.b. plots the platform revenue share, given by the ratio of the cumulative platform earnings to the cumulative creator earnings. The stationary 0.05 value is expected since it is the constant multiplicative factor applied to each reward in the simulation to compute the platform's commission on the reward.

Fig. 2c plots the project-solving efficiency over time, which we define as the ratio of the number of creators who solved at least one project in a hackathon to the total number of creators in that event. In practice, this is the same as the ratio of creators with earnings divided by the total number of creators in the event. For the 50 events (50 times steps) illustrated, we see the solving efficiency fluctuating around 90%.

First, the efficiency fluctuates because the model is stochastic in its implementation (each hackathon has its own set of hackers with random accuracies). Second, the approximate 90% value of Fig. 2c is obtained analytically by realizing that, at each time step, a creator is "exposed" to 10 projects (one for each company). The probability of solving n out of these 10 projects is given by the binomial distribution, where the probability of success p is given by the creator's accuracy, which is uniformly distributed between 0 and 1. Then, the probability of a creator solving at least one project



2.a. Time evolution of platform earnings and creator earnings.	2.b. Time evolution of revenue share.	2.c. Time evolution of Project solving efficiency.

Figure 2: The basic model: time evolution of earnings, revenue share, and project-solving efficiency. Note that a one-time unit is the equivalent of participation in one hackathon.

is (1-probability of solving none):

$$1 - (1 - p)^{10}$$

Hence, this probability's average is

$$1 - \int_0^1 (1 - p)^{10} dp = 1 - \frac{1}{11} \cong 0.91,$$

which is the value around which the plot of Fig. 2c. fluctuates.

As mentioned above, the results presented in Figure 2 analyzed the situation when the reward pool is infinite in size and considered in Figure 2.a the aggregate earnings over all creators. We will now provide some results for the situation when the reward pool is finite.

Recall, from Algorithm 1, that the creator earnings are paid from this pool and that the pool is added 20% of platform earnings each time a project is successfully solved. In this situation, to prevent a negative pool balance, if, at some point in time, the reward to be bestowed to a creator is higher than the pool balance, then this balance is depleted, and the creator earns only a fraction of the eligible reward that the pool balance can afford. This introduces inequality in creator earnings since some hackers may be lucky to find the reward pool replenished while others may find it depleted. To see this effect, we plot in Figures 3 and 4 the earnings for 10 creators for two different reward pool sizes resulting from two simulations. In the simulation of Figure 3, the pool size is 2000, while in that of Figure 4, the pool size is 50. Each curve corresponds to one creator; the accuracy is indicated in the accompanying legend.

As can be seen, the case with the smaller pool size of Fig. 4 shows unequal earnings after 10 hackathons are simulated. For example, the hacker represented by the cyan curve in Fig. 4 was unlucky in finding the pool depleted in many hackathons and ended with earnings of only about 60, despite her or his high accuracy (0.8214).

In order to better understand the inequality in hacker earnings originating from a finite pool size, we ran new simulations of 10 hackathons, but this time with 500 hackers. At the end, i.e., after the 10^{th} hackathon, we computed the Gini coefficient of the hackers' earnings. This is an important consideration as the better distribution of earnings from hackathons is likely to result in higher participation and contribute to the sustainability of the platform.

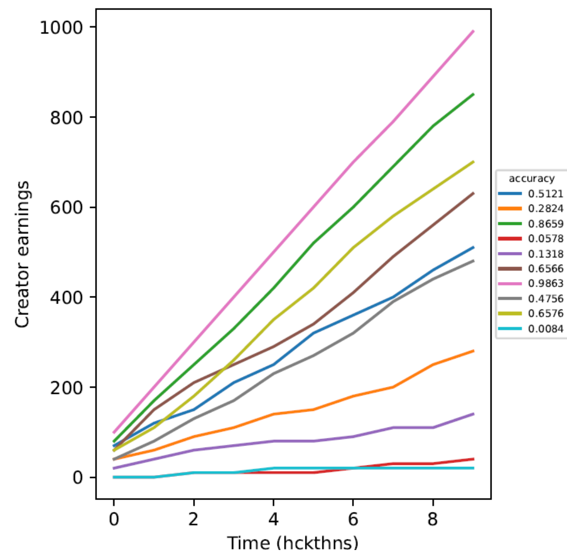


Figure 3: Behavior of creator earnings (segmented by accuracy) over time when a reward pool size is 2000.

We ran the simulations and computed the Gini coefficient for a variable range of pool sizes. The result is plotted in Fig. 5

Recall that the Gini coefficient is a measure of the inequality of a set of data [27]. Figure 5 then shows that large pool sizes result in a low Gini coefficient (i.e., low inequality) of hacker earnings. A very small pool size also results in low inequality of hacker earnings. This is to be expected since when the pool size is very small, all hackers are likely to always find the pool depleted, and thus, all hackers will receive earnings that are approximately equal. However, for intermediate pool sizes, we can see from Figure 5 that the Gini coefficient has a peak. It is for these intermediate pool sizes that, in the simulations, some hackers find the pool sufficiently replenished to earn the eligible reward, while others find it depleted. This is the origin of the high inequality (high Gini coefficient) observed for intermediate pool sizes.

Algorithm 1 Hackathon Processing Loop

Inputs:
Instance of the Platform class
hckrs: List of Creator objects (hackers)
companies: List of Company objects
reward_pool: Current amount of tokens in Platform's reward pool
Outputs:
Updated hckrs: List of Creator objects with rankings (ordered by solved projects and earnings)
Updated companies: List of Company objects reflecting earnings for each hackathon
Updated reward_pool: Reflecting the Platform's remaining reward tokens
Current Hackathon: hckthn (initialized as the first element in hckthns)
Current Project: project (uninitialized)
Current Hacker: hckr (uninitialized)
Steps:
While there are unprocessed hckthns:
Set hckthn to the current element in hckthns.
While there are unprocessed projects (from companies) in the current hckthn:
Set the project to the next project retrieved (from companies).
Shuffle the hckrs list order (to introduce randomization in terms of who is the first to resolve)
For each hckr in hckrs list:
Set hckr to the current element in hckrs.
Attempt Solve Project (probability of success determined by the hckr's accuracy).
Update on Success:
If successful solving:
Decrement company.earnings by project["reward"].
Calculate platform earnings (platform_earnings) as 5% of project["reward"].
Update platform and company earnings (platform.earnings += platform_earnings, company.earnings -= platform_earnings).
Add 20% of platform_earnings to the reward pool (platform.reward_pool += platform_earnings * 0.2).
Distribute rewards to the successful hckr (limited by the pool balance) using platform.distribute_rewards(project["reward"], hckr).
Move to the next hackathon in the list.
End Simulation

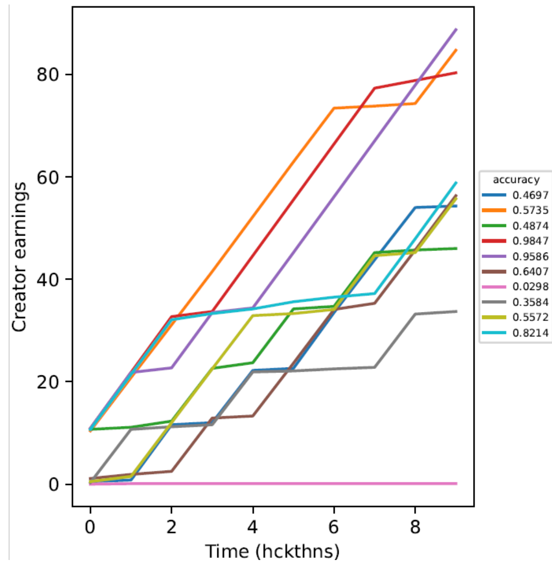


Figure 4: Behavior of creator earnings (segmented by accuracy) over time when a reward pool size is 50.

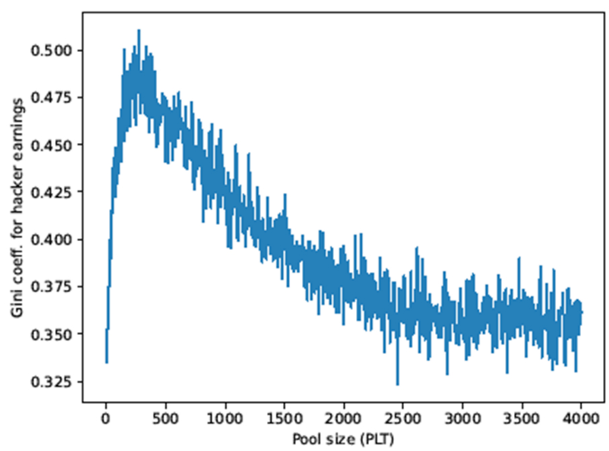


Figure 5: Gini coefficient of hackers' earnings as a function of pool size.

Now, consider the following improvement to the model: As previously mentioned, the model assumes that the hackers' accuracies are random uniform between 0 and 1. This assumption can be improved by realizing that, during the 10 hackathons, the 500 hackers can enhance their accuracy by a specific amount with each successful project completion. A simple way to account for this learning factor would be to assume that each time a project is solved by a hacker, that hacker's accuracy increases by some additive factor, let's say α, meaning that the hacker's accuracy is updated as follows: $accuracy \longleftarrow \min(1, accuracy + \alpha)$.

To illustrate the effect of including the additive learning factor α, we again plot the Gini coefficient of hacker earnings as a function of the pool size in Figure 6. Three curves are plotted, each corresponding to a value of α: 0.001 (green), 0.01 (red), or 0.1 (cyan). As can be seen, the higher the learning factor, the lower the Gini coefficient and, thus, the lower the inequality of hacker earnings. This is expected, given that a high α will result in a significant number of hackers achieving an accuracy close to 1 after 10 hackathons have been completed, thereby creating a more equal situation than if their accuracies were distributed uniformly between 0 and 1 (see the cyan curve of Fig. 6). Conversely, for a low α (see the green curve of Fig. 6), only a small subset of hackers will complete the 10 hackathons with an accuracy of 1 (or a value close to 1). The remaining hackers will still improve their accuracy, but not to the

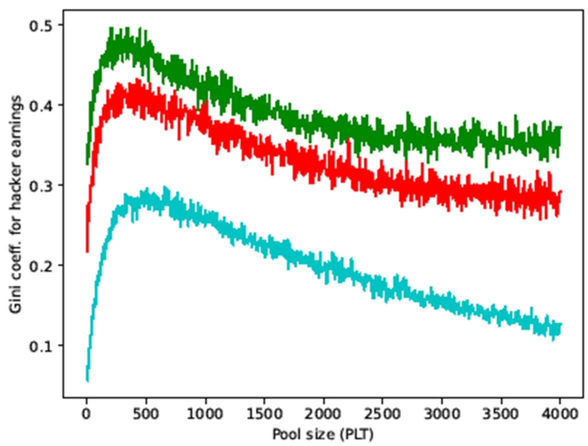


Figure 6: Gini coefficient of hackers' earnings as a function of pool size for three values of the learning factor α: 0.001 (green), 0.01 (red), and 0.1 (cyan).

extent that they will achieve an accuracy level of 1. Consequently, a low α is expected to result in a situation where inequality is greater than that observed with a high α.

4 CONCLUSION

The proposed agent-based model is a tool for platform owners and researchers to study the effect of strategic choices on platform outcomes. In this case, we show how the variation of one of the tokenomics variables (rewards pool size) together with the choice of projects (which are associated with learning) could have very different results in terms of the individual earnings, the distribution of earnings (equity), but also in individual learning and problem-solving accuracy. In turn, these choices influence how problem-solvers (creators) perceive the digital platform and hackathons and whether they would return to contribute or recommend to others to join. As hackathon platforms depend on the creators, their continuous participation and growth in numbers are critical for platform sustainability.

While this study shows the potential of agent-based modeling as a decision-making aid for platform design and incentives, blockchain has a critical role in enabling the sustainability of this system. Blockchain, through smart contracts and tokens, allows for creating a trustworthy rewards system. Not only does it allow each contribution to be registered and verifiable, but blockchain technology also expands the frontier of possible. In particular, as platforms launch their tokens, they gain flexibility in defining the rewards pool size but also add to the expectations of future earning as the token's popularity grows or inflationary mechanisms activate on top of the token demand growth. That entire expectation hinges on the popularity of the digital platform for hackathons and the demand for the tokens. Therefore, extensive simulations, as proposed in this work, are an indispensable part of managerial action. While the models cannot guarantee a perfect system or absolute success in the market, they can help improve the experience of hackathon platforms.

The ABM could also assist in exploring mechanisms for revisiting past unsuccessful projects. By incorporating metrics such as project code quality, community reception (if applicable), and alignment with emerging trends, the model could identify projects with potential despite not initially receiving recognition. The platform could then leverage its matching capabilities to connect these projects with interested companies facing similar challenges in subsequent hackathons. This could potentially result in a ”second chance” for valuable solutions and incentivize continued participation from talented creators who might have felt discouraged by not winning in the past iteration.

There are two main avenues of future research. One is to further extend the model, for example, by studying the impact of the reputation system on Creators (R). Finding relevant metrics is also part of this avenue, which helps include these metrics into a reputation score or other system elements. Additionally, exploring varying risk tolerance levels for companies, the heterogeneity of projects, and incorporating budget constraints into project selection criteria would provide a more nuanced representation of real-world client behavior. The model can also be expanded to encompass a broader range of agent interactions. This could include the ability for companies to directly interact with creators based on their skills and reputation or for creators to collaborate on projects. By systematically incorporating these elements and validating the model with real-world data, this ABM can become a valuable tool for designing and optimizing efficient, rewarding, and sustainable tokenized freelancing platforms. The second avenue is the experimentation. Once the models are built and ideally tested in predicting the observed behaviors, their value is best tested in creating interventions and comparing the model outcomes with the real-world systems. When digital platforms integrate blockchain technologies, the recorded data allow for the verification of the extent to which the predictions correspond to reality.

Acknowledgments

This work was financially supported by Project Blockchain.PT – Decentralize Portugal with Blockchain Agenda, (Project no 51), WP5: New Data and Knowledge Economy, Call no 02/C05-i01.01/2022, funded by the Portuguese Recovery and Resillience Program (PPR), The Portuguese Republic and The European Union (EU) under the framework of Next Generation EU Program.

References

[1] Flus, M. and Hurst, A. (2021). Design at hackathons: new opportunities for design research. Design Science, 7. https://doi.org/10.1017/dsj.2021.1
[2] Garcia, M. (2023). Fostering an innovation culture in the education sector: a scoping review and bibliometric analysis of hackathon research. Innovative Higher Education, 48(4), 739-762. https://doi.org/10.1007/s10755-023-09651-y
[3] Tan, R., Wu, D., Day, S., Zhao, Y., Larson, H., Sylvia, S., . . . & Tucker, J. (2022). Digital approaches to enhancing community engagement in clinical trials. NPJ Digital Medicine, 5(1). https://doi.org/10.1038/s41746-022-00581-1
[4] Martinec, T. (2024). Virtual design hackathons: a data collection framework. Proceedings of the Design Society, 4, 45-54. https://doi.org/10.1017/pds.2024.7
[5] Mendes, W., Albert, R., Tillo, T., Pinto, G., Gama, K., & Nolte, A. (2022). Socio-technical constraints and affordances of virtual collaboration – a study of four online hackathons. Proc. ACM Hum. Comput. Interact. 6, CSCW2, Article 330 (November 2022), 32 pages. https://doi.org/10.1145/3555221
[6] Taly, A., Poulain, P., Dufour, H., & Bertrand, M. (2021). Ten simple rules for implementing a flipped classroom.. https://doi.org/10.20944/preprints202007.0030.

v2

[7] Obwegeser, N. and Müller, S. (2018). Innovation and public procurement: terminology, concepts, and applications. Technovation, 74-75, 1-17. https://doi.org/10.1016/j.technovation.2018.02.015

[8] Khoshavi, N., Tristani, G., & Sargolzaei, A. (2021). Blockchain applications to improve operation and security of transportation systems: a survey. Electronics, 10(5), 629. https://doi.org/10.3390/electronics10050629

[9] Behl, A., Pereira, V., Nigam, A., Wamba, S., & Sindhwani, R. (2023). Knowledge development in non-fungible tokens (nft): a scoping review. Journal of Knowledge Management, 28(1), 232-267. https://doi.org/10.1108/jkm-12-2022-0937

[10] Hope, A., D'Ignazio, C., Hoy, J., Michelson, R., Roberts, J., Krontiris, K., . . . & Zuckerman, E. (2019). Hackathons as participatory design. Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems. https://doi.org/10.1145/3290605.3300291

[11] Kitsios, F. and Kamariotou, M. (2019). Beyond open data hackathons: exploring digital innovation success. Information, 10(7), 235. https://doi.org/10.3390/info10070235

[12] Willian Mougayar. 2016. The Business Blockchain: Promise, Practice, and Application of the Next Internet Technology. John Wiley & Sons. New Jersey, USA.

[13] Aneesh Zutshi, Antonio Grilo, Tahereh Nodehi. 2021. The value proposition of blockchain technologies and its impact on Digital Platforms. Computers & Industrial Engineering. 155, 107187 (February 2021), 1-17. https://doi.org/10.1016/j.cie.2021.107187

[14] Romildo Silva, Rui Pedro Marques, Helena Inácio. 2024. A design for tokenization in governmental investment. International Journal of Accounting & Information Management. 32, 1 (December 2023),19-39. https://doi.org/10.1108/IJAIM-03-2023-0070

[15] Lin William Cong, Ye Li, and Neng Wang. 2022. Token-based platform finance. Journal of Financial Economics. 144, 3 (October 2021), 972-991. https://doi.org/10.1016/j.jfineco.2021.10.002

[16] Michael Sokin and Wei Xiong. (2023). Decentralization through Tokenization. The Journal of Finance. LXXVIII,1 (February 2023), 247-299.

[17] John Sterman. 2017. Business dynamics: systems thinking and modeling for a complex world. McGraw-Hill. New York, USA.

[18] Sarah Otto and Troy Day. 2007. A Biologist's Guide to Mathematical Modeling in Ecology and Evolution. Princeton University Press. Princeton, USA. https://doi.org/10.1515/9781400840915

[19] Nicholas Britton. 2012. Essential mathematical biology. Springer Science & Business Media. London, UK.

[20] Mark Newman. 2010. Networks: An Introduction. Oxford University Press. Oxford, UK.

[21] Alberto-Laszlo Barabasi. 2002. Linked: How Everything Is Connected to Everything Else and What It Means for Business, Science, and Everyday Life. Perseus Books Group. Cambridge, USA.

[22] Sergey Dorogovtsev and José Fernando Ferreira Mendes. 2022. The Nature of Complex Networks. Oxford University Press. Oxford, UK.

[23] Eric Bonabeau. 2002. Agent-based modeling: Methods and techniques for simulating human systems. In Proceedings of the National Academy of Sciences, 99, 3, 7280-7287. https://doi.org/10.1073/pnas.082080899

[24] Stephan Mertens and Cristopher Moore. 2011. The Nature of Computation. Oxford University Press. Oxford, UK.

[25] Christoph Mueller-Bloch,Jonas Valbjørn Andersen,,Jason Spasovski, and Jungpil Hahn. 2022. Understanding decentralization of decision-making power in proof-of-stake blockchains: An agent-based simulation approach. International Journal of Electronic Governance, Research and Applications. 17,4 (September 2022), 381-400. https://doi.org/10.1080/0960085X.2022.2125840

[26] Joshua Epstein. 2006. Generative social science: How agent-based modeling can help us understand human behavior. Princeton University Press. Princeton, USA.

[27] A. Atkinson, "On the measurement of inequality", Journal of Economic Theory, vol. 2, no. 3, p. 244-263, 1970. https://doi.org/10.1016/0022-0531(70)90039-6