sustainability

Article

Breaking Triopoly to Achieve Sustainable Smart Digital Infrastructure Based on Open-Source Diffusion Using Government–Platform–User Evolutionary Game

Tao Li [1,2,3], Junlin Zhu [1,2,*], Jianqiang Luo [4,*], Chaonan Yi [1,2] and Baoqing Zhu [1,2]

1 School of Intellectual Property, Nanjing University of Science and Technology, No. 200 Xiaolingwei Street, Xuanwu District, Nanjing 210094, China; taolee@njust.edu.cn (T.L.); njlgdx10095@njust.edu.cn (C.Y.); 222119010108@njust.edu.cn (B.Z.)
2 Centre for Innovation and Development, Nanjing University of Science and Technology, Nanjing 210094, China
3 School of Business, Xianda College of Economics & Humanities Shanghai International Studies University, No. 390 Dong Tiyuhui Rd., Hongkou District, Shanghai 200083, China
4 School of Management, Jiangsu University, No. 301 Xue Fu Rd., Zhenjiang 212013, China
* Correspondence: jl_zhu@njust.edu.cn (J.Z.); ljq2809@163.com (J.L.)

Abstract: Technological innovations, including the Internet of Things (IoT) and machine learning, have facilitated the emergence of autonomous systems, promoting triple bottom line (TBL) sustainability. However, the prevalent triopoly of Android, iOS, and Windows introduces substantial obstacles for smart device manufacturers in pursuit of independent innovation. This research endeavors to elucidate how open-source operating systems can counteract this triopoly and catalyze sustainable digital development. Utilizing evolutionary game theory, we scrutinize the interplay among governments, platforms, and users in championing open-source diffusion. Our analysis unveils two potent evolutionary strategies—incentivized engagement and disengagement—that notably expedite open-source diffusion and attenuate software supply chain risks affiliated with the Android–iOS–Windows triopoly (results). Consequently, this research highlights the critical role of augmenting stakeholder collaboration and bolstering platform reputation in propelling open-source diffusion, thereby providing valuable theoretical insights and practical guidance for the sustainable advancement of smart digital infrastructure.

Keywords: open-source diffusion; sustainable smart digital infrastructure; open-source operating system; evolutionary game; open innovation

Citation: Li, T.; Zhu, J.; Luo, J.; Yi, C.; Zhu, B. Breaking Triopoly to Achieve Sustainable Smart Digital Infrastructure Based on Open-Source Diffusion Using Government–Platform–User Evolutionary Game. *Sustainability* **2023**, *15*, 14412. https://doi.org/10.3390/su151914412

Academic Editors: Rashid Mehmood, Tan Yigitcanlar and Juan M. Corchado

Received: 29 August 2023
Revised: 27 September 2023
Accepted: 28 September 2023
Published: 1 October 2023

1. Introduction

With the rising complexity of contemporary smart city development and deteriorating environmental and economic sustainability, the triple bottom line (TBL), i.e., social, environmental, and economic sustainability, calls for the engagement of open innovation-style technologies [1]. Open source is an open innovation model in the digital era [2]. Open-source diffusion constitutes an information propagation paradigm, anchored on the tenets of unrestricted access, collective sharing, and amendment of source codes, knowledge, and information resources. Initially rooted in the field of computer software development, it encompasses not only the application of open-source software and open-source hardware, but also the sharing and utilization of open-source resources such as open-source data; it embodies a community-powered innovation model. The characteristics of open-source diffusion, such as openness, sharing, transparency, and customisability [3], fit with the idea of green AI aiming at efficient, sustainable and equitable development of smart cities and future technologies [4,5], and it can provide a supportive and facilitating mechanism for autonomous systems and smart digital sustainable implementations [6].

The development of open-source operating systems, as a crucial vehicle for open-source diffusion, contributes to the diversification of intelligent technologies and enables market freedom of choice [7–10]. In competition among open-source operating systems (Table 1), kernel update and optimization of an operating system is a core task of the open-source platform [11,12]. Kernel is the core of an open-source operating system, which controls the operation of computer hardware and provides services for applications. Open-source software includes the kernel of the open-source operating system, as well as other application programs, tools, and libraries, which constitute an open-source software ecosystem [13]. The source code of the operating system kernel is open for sharing [14], and people can use, modify, and disseminate it freely. Open-source operating systems have diverse brands and business models [15], and the license adopted also affects its development, distribution, and use [16].

Table 1. Open Source Operating System Competitive Landscape.

Brand	Release Version	Kernel	License	Description	Area
Linux	Ubuntu Debian CentOS Red Hat	Linux	GPL/MIT	One of the most popular open-source operating systems, widely used in servers, workstations and personal computers, with a large community of developers	Global
Kylin	NeoKylin KylinOS	Linux	GPL/MIT	Native Chinese operating system developed for China, mainly used by the Chinese government and for enterprise information construction.	China
UOS	UOS Desktop UOS Server	Linux	GPL/MIT	China's self-developed enterprise-class operating system, designed to replace foreign operating systems and improve information security and autonomous control.	China
FreeBSD	FreeBSD OpenBSD NetBSD	BSD	BSD	A Unix-like operating system, known for reliability, performance and security; it is widely used in servers, embedded, desktop systems and routers.	Global
OpenSolaris	Solaris	Solaris	CDDL	Known for being the world's most advanced file system and networking protocol, widely used in servers, desktop systems and virtualized environments, but discontinued by Oracle for maintenance.	Global
Android	Android	Linux	Apache	A mobile device operating system with cell phones and tablets as the main target; based on Linux kernel and the open-source project AOSP development.	Global
Chrome OS	Chrome OS	Linux	Chromium	Google's operating system based on Linux kernel and the Chrome browser; mainly used for cloud computing and lightweight devices.	Global
ReactOS	ReactOS	NT	GPL	Open source, Windows-compatible operating system designed to replace Windows and provide a high degree of compatibility and stability.	Global

Table 1. *Cont.*

Brand	Release Version	Kernel	License	Description	Area
Sailfish OS	Sailfish OS	Linux	MPL	Open-source mobile operating system from Finland, supporting Android applications; it is known for security, privacy and personalization.	Finland
Raspberry Pi OS	Raspberry Pi OS	Linux	GPL/LGPL	Debian-based operating system designed for Raspberry Pi; it is designed to provide a clean and fun environment for learning, exploration and innovation.	U.K.
Fedora	Fedora	Linux	GPL/MIT	Community-driven, free Linux operating system designed to experiment with new technologies, improve the developer experience, and deliver the latest packages as an upstream version of Red Hat Enterprise Linux.	Global
HarmonyOS	HarmonyOS	Microkernel	Apache	A distributed operating system independently developed by Huawei, it is designed to achieve cross-terminal multi-terminal collaborative operation, supporting smartphones, smart wear, car systems, etc.	China

Currently, the Android–iOS–Windows triple oligarchy of operating systems needs to be broken [17,18], which not only reduces user dependency risks and introduces more competitors, igniting greater innovation vitality, but also holds significant importance in mitigating the risk of supply chain disruptions. In parallel, the lack of resources, developers or maintainers in the open source community, or unhealthy software industry structures and trade conflicts can lead to supply chain disruption risks for open-source operating system software. Therefore, intelligent device manufacturers have started actively developing their own open-source operating systems to counter the threat of proprietary software monopolies and technology disruptions, also adopting incentive mechanisms such as software policies [19], open data [20], infrastructure [21,22], talent cultivation [23], and global collaboration [21,22] to further promote open-source diffusion [24].

To mitigate the risks associated with the supply chain of the Android–iOS–Windows triple oligarchy, open-source platforms can endeavour to expand their user communities and attract more developers to participate. Additionally, governments can enhance regulations, incentivize maintenance efforts, and allocate public resources and funding to foster the diffusion of open-source technologies and contribute to the sustainability, fairness, and efficiency of smart digital technologies. Therefore, this research is grounded in the theory of open-source diffusion and utilizes an evolutionary game approach to investigate the evolving dynamics among three key stakeholders: open-source platforms, users, and governments, within the realm of open-source diffusion. Finally, this paper proposes targeted recommendations to mitigate risk of supply chain disruptions through open-source diffusion, thereby fostering sustainable smart digital infrastructure.

This study demonstrates innovation in several aspects. Firstly, it offers a novel theoretical interpretation of how open-source diffusion becomes a pivotal factor in supporting the sustainable development of intelligent digital infrastructure. Furthermore, it presents fresh perspectives and strategies on how to disrupt the existing triopoly of the Android–iOS–Windows "Big Three" through open-source diffusion. In contrast to previous studies concentrated solely on individual stakeholders, this research adopts the evolutionary game model to encompass multiple stakeholders, such as the government, the platform, and the user. By including a diverse range of actors, this study offers a more holistic understanding of the complexities involved in open-source diffusion. It particularly focuses on the role of government in driving open-source diffusion and fostering the sustainability of smart digital infrastructure, providing effective policy recommendations in this regard.

The structure of this article is as follows: Section 1 explains the research background, raises research questions, introduces the methods, and describes the contributions. Section 2 discusses recent related work that support our study. In Section 3, we introduce the evolutional game, solve equilibriums, perform simulations and analyse the results. In Section 4, we discuss the findings, and summarize the study in Section 5.

2. Related Works

2.1. Open Source in Smart Digital Technologies

Smart digital technologies become common solutions to urban crises associated with the climate, epidemics, natural disasters, and socio-economic factors [25,26] concerning the triple bottom line (TBL), i.e., social, environmental, and economic sustainability [1]. Artificial intelligence is rapidly becoming a key element of smart cities [27,28], helping to improve efficiency and automation [29]. Such technology poses significant risks of privacy violations and disruption through opaque decision-making processes [30]. Emerging challenges, including massive data, heterogeneities, complex dependencies, distributed storage and computing, and data [31], are open issues that need to be confronted by smart digital technologies.

The triopoly shaped by Android–iOS–Windows has brought about a technological ecological monopoly, limiting competition and innovation [17], thus leading to a lack of diversity and flexibility in the development of smart digital technologies and smart cities [18] as well as restricting consumer choice and innovation [32]. Open-source operating systems, such as Debian 12 [33], Ubuntu 23 [34], and HarmonyOS 4 [34], and open-source software, such as PyTorch 2 [35] and SciPy 1 [36], promote technology sharing and cooperation, break monopolies, establish open technology platforms [37], build collaborative ecosystems, and cultivate an open innovation culture [38]. Thus, smart digital technologies and the sustainable development of smart cities complement open source and open innovation [39], and together they promote the sustainable development of society [40].

Open innovation and open source are intricately connected. As outlined by Chesbrough [41,42], at the heart of open innovation lies the pursuit of external innovation resources from both within and outside the organization in order to generate value [43], enhance the efficiency and quality of innovation [44], spur technological transformation [45], and drive business model innovation [46]. Additionally, it promotes international collaboration and knowledge sharing [47,48], facilitates the reconfiguration and optimization of intellectual property and industrial chains [49], and fosters the overall development of industries and economies [50,51]. By adopting the paradigm of open innovation, open source is an internet-based collaborative model that pools efforts of crowd intelligence through the open sharing of knowledge and technical resources [52] to improve efficiency and quality of innovation [53]; this advocates free, shared, and co-creation [3], and it effectively addresses technological inequities and privacy violations [54]. Open-source operating systems promote technological security [55], technological pluralism [56] and smart digital infrastructure [57]. Open source as public goods expanding business models, based on its tacit knowledge [58], can be the activation engine for innovative regional development [59,60] and smart technologies [4,40,61,62]. During the evolution of open source [63], attention concerning the ecosystem radiates from hardware and software to their developers, users, communities, platforms and other relevant organizations and government departments [64], achieving fast, flexible, and secure application development through collaborations [65].

2.2. Infrastructure Risk Mitigation

With the advancement of information technology and digital transformation, an increasing number of enterprises now rely on software operating systems, and the risk of software supply chain disruption has gradually received more attention [66]. The risk of software supply chain disruption typically includes software vulnerabilities and malicious code, supplier bankruptcy, cyberattacks, and global crises, resulting in major impacts on a company's business operations and data security [67]. Enterprises need to take effective measures to manage the risks in software supply chain disruption [68], establish socio-technical frameworks [69], and reduce the impact of disruptions. Open source helps enterprises mitigate the risks of software supply chain disruption [70,71]. For instance, enterprises can use open-source software vulnerability scanners to scan for vulnerabilities in software systems and detect and repair potential issues in a timely manner [72]. Furthermore, enterprises can use open-source software supply chain security auditing tools to check source codes and assess whether security practices and processes of the suppliers comply with standards [73]. Long-term cooperation with software suppliers can also reduce the occurrence and impact of software supply chain disruption [74]. When seeking software suppliers, enterprises should focus on their reliability, technical capabilities, and security measures, establishing long-term cooperative relationships. Moreover, enterprises should work with software suppliers to explore novel security solutions, establish mutually trusted cooperative relationships, and jointly address the risk of software supply chain disruption [75].

Open source has the advantage of coping with the risk of software supply disruptions. However, open source does not reach a wide enough audience, and its low market share is major disadvantages [76,77]. Thus, the mechanism of open-source diffusion has become an urgent issue needing to be addressed.

2.3. Open-Source Diffusion

Joseph Schumpeter was the first scholar to discuss the issues of technological diffusion and product diffusion [78]. He believed technological diffusion was the process through which new technologies gradually spread from research centres to peripheral areas. In this process, the flow of technology from innovators to imitators was viewed as a kind of knowledge "penetration", which promoted productivity growth. Rapid diffusion of technology could greatly accelerate economic growth. Product diffusion was a special form of technological diffusion, referring to the process of expanding products produced originally in one country or region to other countries and regions through exportation, franchising, and other means. Product diffusion could bring wider markets and higher profits, further promoting economic growth.

The scope of technology diffusion mainly includes the domestic and global markets, intra- and inter-enterprise markets, and government markets [79]. The models for calculating technology diffusion include the innovation diffusion model, incentive model, and contagion model [80]. In practice, technology diffusion connects product diffusion. Product diffusion refers to the process of introducing and promoting new products in the market, including market demand, market segmentation, and promotion strategies [81].

Open-source software is diffused at the level of artificial artifacts [82,83]. The essence of open-source diffusion is the shared spreading of source code and technical documentation as well as asynchronous participatory innovative iterations [84]. Community-driven operation facilitates rapid software diffusion [85], and open-source software developers interact with users on community platforms to acquire user needs and proactively address relevant user issues, thus providing technical support and upgrade services to increase user retention and loyalty. Industry applications promote open-source diffusion [86] by integrating open-source software into vertical application systems; it enables vertical industries and application areas, increases software market value and demand, and achieves rapid diffusion and profit. In social network operation [87], open source software can use social media and other network platforms for community and brand promotion, actively expand user networks and increase software awareness.

In summary, open source has characteristics of openness and sharing, and is a paradigm of open innovation. Its diffusion model has unique methods. Open-source diffusion is expected to further promote smart digital infrastructure and sustainability as well as mitigate software supply chain risks.

3. Method

In open source, operating system software is fast becoming the mainstream, mainly used in industries such as the internet, cloud computing, security, and communications. Compared to traditional proprietary operating system software, open-source operating system software has the advantages of low cost, high flexibility, and good security, and its application has gradually expanded to fields such as industry, healthcare, and finance. However, the development of new releases based on various kernels is also influenced by factors such as the technical diffusion environment and policy support. Policies should not only be beneficial to users but also foster the promotion and use of open-source operating system software.

Therefore, we use the evolutionary game framework to study decision factors affecting the promotion of open-source operating system software, to find evolutionary paths, and to discuss the theoretical and practical significance of the results.

In Figure 1, a brief summary diagram of the entire analytical workflow for Section 3 is presented. This figure details the technical results of each key work node as well as the corresponding diagrams and formulas for the subsequent subsections. This figure helps the reader to quickly capture the main flow of the subsequent analysed content, and it allows the reader to locate the corresponding diagrams and related information quickly.

3.1. Premise

The underlying assumptions are as follows: the government, open-source platform and user are players (participants) in this study, conforming to limited rationality and making decisions with the goal of maximizing benefits [88]. The randomness of the strategic choices in decision making is expressed as probabilities in game theory, corresponding to the level of willingness of the players in a real situation [89,90].

The PAPI components are as follows: To present the game, we use the Players + Actions + Payoffs + Information (PAPI) framework [91], which reflects the dynamic interaction between players' decisions based on the information they observe and behaviours of others. Evolution is a long-term process with players' decisions adjusted over time, and they influence each other.

P (Players): players involved in open-source diffusion include government (denoted as G), open-source platform (denoted as P), and user (denoted as U).

A (Actions): government plays a guiding role by fostering the promotion of domestic open-source software through policies, regulations, and funding support, improving the software quality and security and facilitating the development of the domestic IT industry.

Government implements strategies: {Incentive (IC), Not Incentive (NIC)}.

As an open-source software provider, an open-source platform needs to consider the development of technology and market demand as well as promote diffusion and popularity of their software to increase the number of users and market share.

Platform has strategies: {Iterate(IR), Not Iterate(NIR)}.

User needs to balance software quality and migration costs in choosing between open source and proprietary software.

User response strategies: {Feedback(FB), Not Feedback(NFB)}.

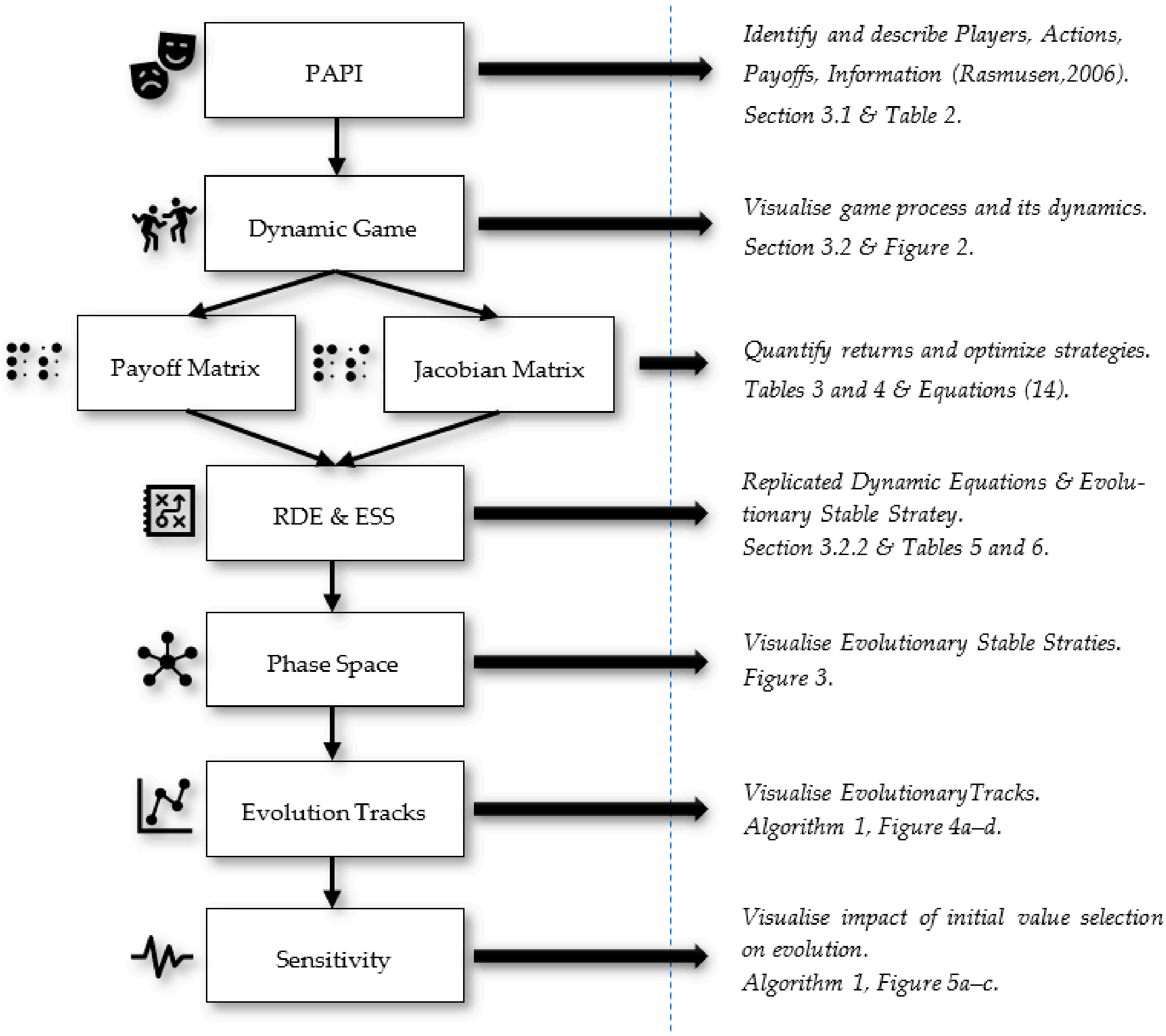


Figure 1. Analytical workflow roadmap [91].

P (Payoffs): The payoffs for players in the evolutionary game are measured in terms of their total return from the respective strategies. For simplicity, we model interactions by providing a payoffs matrix under the evolutionary game model. The process of equilibrating the ESSs based on replicated dynamic equations (RDEs) are then given sequentially in Section 3.3.

I (Information): In the evolutionary game, the government, open-source platform, and user need to balance their interests through cooperative negotiation, jointly promote the development and popularity of open-source software, and achieve a win–win situation. In short, from the government's perspective, on the platform side, the government incentives promote the diffusion behaviour of open-source platforms, prompting open-source platforms to actively seek potential user groups, actively collect user feedback, and actively innovate and optimize open-source operating system software. On the user side, government incentives guide the user to actively accept the diffusion from the open-source platform, actively submit feedback on software usage, and form a loop with the open innovation of open-source platforms to create synergy.

A list of symbols denoting the variables in the following subsections is shown in Table 2.

Table 2. List of symbols.

Symbol	Type	Descriptions
$\alpha \in [0, 1]$	probabilistic	Probability of government-imposed incentives
$\beta \in [0, 1]$	probabilistic	Probability of the open-source platform to enforce diffusion
$\gamma \in [0, 1]$	probabilistic	Probability of user-implemented feedback
$C_1 \in [0, 1]$	economic	Costs incurred by government incentives for open-source platforms
$C_2 \in [0, 1]$	economic	Costs incurred by government incentives for users
$p \in [0, 1]$	proportionate	Ratio of government non-incentives to incentive-generated benefits
$C_0 \in [0, 1]$	economic	Costs invested by open-source platforms and users for open-source diffusion in the absence of government incentives
$\Delta C \in [0,1]$	economic	Reduction in the costs invested by open-source platforms and users for open-source diffusion in the presence of government incentives
$\theta \in [0, 1]$	proportionate	Cost-sharing ratio between open-source platforms and users
$R_1 \in [0, 1]$	economic	Benefits generated by government incentives for open-source diffusion
$R_2 \in [0, 1]$	economic	User benefits in the initial state
$R_3 \in [0, 1]$	economic	Benefits of open-source platforms in the initial state
$R \in [0, 1]$	economic	Additional benefits for open source platform and user based on open-source diffusion
$e \in [0,1]$	proportionate	Allocation of additional benefits to open source platforms and users based on open-source diffusion
$L_1 \in [0, 1]$	economic	Benefits to users from feedback without government incentives
$L_2 \in [0, 1]$	economic	Benefits gained from proactive innovation iterations of open-source platforms without government incentives
$\tau_1 \in [0, 1]$	economic	Transfer payments for losses to open-source platforms without feedback from users under government incentives
$\tau_2 \in [0, 1]$	economic	Transfer payments for losses to users from open-source platforms without innovation iterations under government incentives

Note: The domain of definition of the above probabilistic and proportionate parameters is between 0 and 1, in line with probability theory. The remaining economic implication-type parameters can be normalized and thus converted to the [0, 1] interval.

3.2. Model

3.2.1. Dynamic Games and Payoff Matrices

Dynamic game diagrams are used to describe the decisions of players in the evolutionary game and clearly show forward-looking strategies and outcomes in different decision sequences. According to the discussion in the previous section, the dynamic process of the game between open-source platforms, users and the government is shown by following the G-P-U (Government–Platform–User) dynamic game diagram (Figure 2).

According to the above dynamic game process, the three players implement strategies successively, and the following payoff matrices (Tables 3 and 4) are obtained using the symbolic information of Table 2.

The above evolutionary game payoff matrices (Tables 3 and 4) reflect the benefits or costs corresponding to the decisions. Analysing this allows us to deduce the stable strategies in the evolutionary game. The Evolutionary Stable Strategy (ESS) refers to when all players' strategies are optimized and no alternative better strategies exist. Therefore, calculating the ESS is crucial for understanding in the G-P-U evolutionary game and predicting changes. In the following subsection, we deduce the ESS from the payoffs matrix.

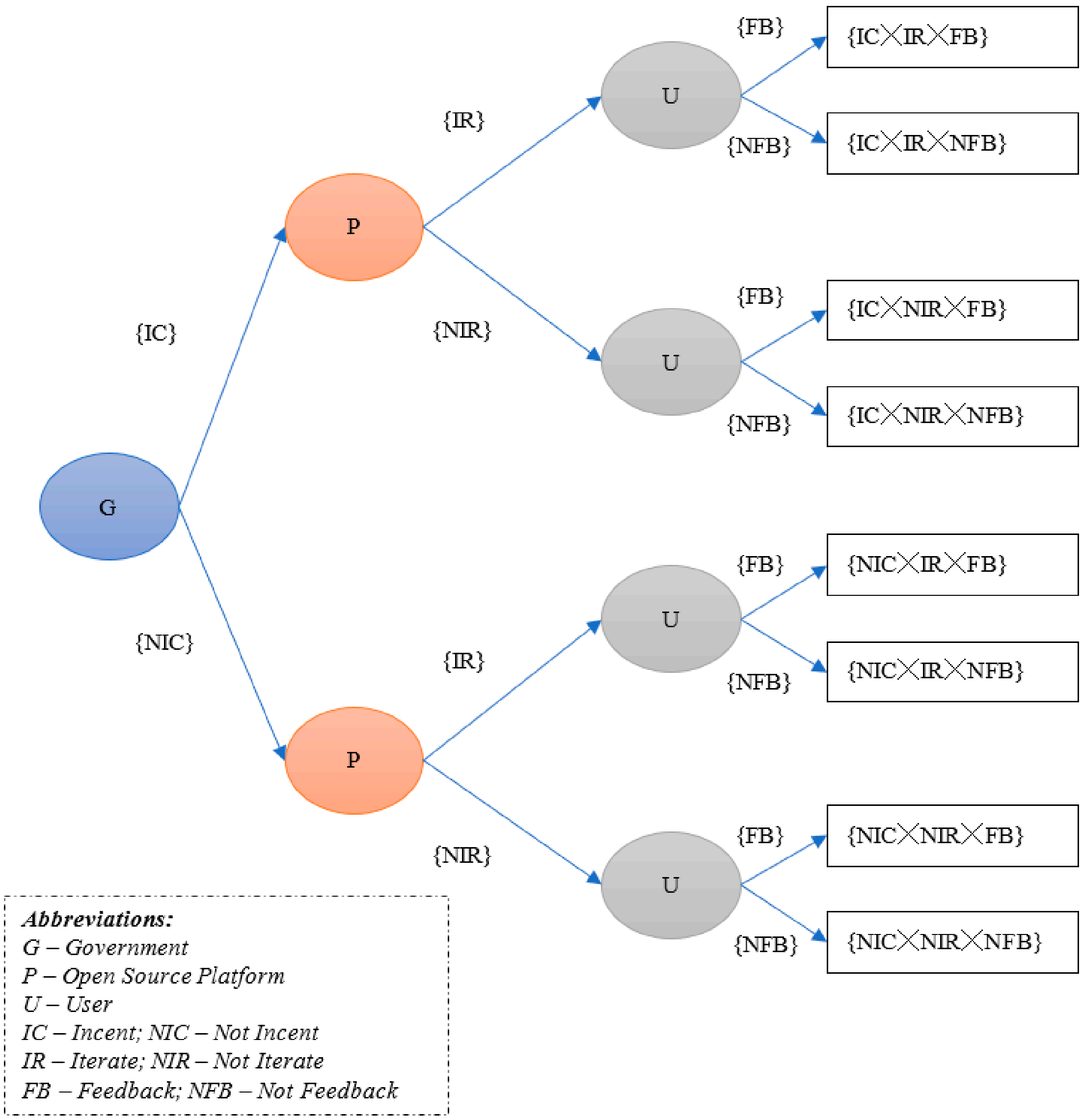


Figure 2. G-P-U dynamic game process.

Table 3. Payoff matrix for the G-P-U OSOSS proliferation evolutionary game (a).

		P {Iterate}	
		U	
		{Feedback}	{Not Feedback}
G	{Incent}	$R_1 - C_1 - C_2;$ $R_2 + \alpha R - \theta(C_0 - \Delta C);$ $R_3 + (1-\alpha)R - (1-\theta)(C_0 - \Delta C) + C_2;$	$R_1 - C_1 - C_2;$ $R_2 + L_1 - \tau_1;$ $R_3 - (1-\theta)(C_0 - \Delta C) + \tau_1 + C_2;$
	{Not Incent}	$pR_1;$ $R_2 + eR - \theta C_0;$ $R_3 + (1-e)R - (1-\theta)C_0;$	$pR_1;$ $R_2 - \tau_1 + L_1;$ $R_3 - (1-\theta)C_0 + \tau_1;$

Note for Abbreviations: G: government; **P**: OSOSS platform; **U**: user.

Table 4. Payoff matrix for the G-P-U OSOSS proliferation evolutionary game (b).

		P {Not Iterate}	
		U	
		{Feedback}	{Not Feedback}
G	{Incent}	$R_1 - C_1$; $R_2 - \theta(C_0 - \Delta C) + \tau_2$; $R_3 - \tau_2 + L_2$;	$R_1 - C_1$; R_2; R_3;
	{Not Incent}	pR_1; $R_2 - sC_0 + \tau_2$; $R_3 - \tau_2 + L_2$;	pR_1; R_2; R_3;

Note for Abbreviations: G: government; **P**: OSOSS platform; **U**: user.

3.2.2. Evolutionary Stable Strategies

The expected return of the government {Incent} strategy is obtained in the following equation:

$$\varphi(\alpha) = \beta[\gamma(R_1 - C_1 - C_2) + (1-\gamma)(R_1 - C_1)] + (1-\beta)[\gamma(R_1 - C_1 - C_2) + (1-\gamma)(R_1 - C_1)] \quad (1)$$

The expected return of the government {Not Incent} strategy is obtained in the following equation:

$$U_{12} = \beta[\gamma p R_1 + (1-\gamma)pR_1] + (1-\beta)[\gamma p R_1 + (1-\gamma)pR_1] \quad (2)$$

The expected return to the government is obtained from Equations (1) and (2).

$$U_1 = \alpha U_{11} + (1-\alpha)U_{12} \quad (3)$$

The government's Dynamic Replication Equation is a derivative of time t.

$$\begin{aligned} H_1 &= \frac{d\alpha}{dt} = \alpha(1-\alpha)(U_{11} - U_{12}) \\ &= \alpha(1-\alpha)\{\beta\gamma[(1-p)R_1 - C_1 - C_2] + \beta(1-\gamma)[(1-p)R_1 - C_1] + (1-\beta)\gamma[(1-p)R_1 - C_1 - C_2] \\ &\quad + (1-\beta)(1-\gamma)[(1-p)R_1 - C_1]\} \\ &= \alpha(1-\alpha)[(1-p)R_1 - C_1 - \beta C_2] \end{aligned} \quad (4)$$

The expected return of the open-source platform {Iterate} strategy is obtained in the following equation:

$$\begin{aligned} U_{21} = {} & \alpha\gamma[R_3 + (1-e)R - (1-\theta)(C_0 - \Delta C) + C_2] + \alpha(1-\gamma)[R_3 - (1-\theta)(C_0 - \Delta C) + \tau_1 + C_2] \\ & + (1-\alpha)\gamma[R_3 + (1-e)R - (1-s)C_0] + (1-\alpha)(1-\gamma)[R_3 - (1-\theta)C_0 + \tau_1] \end{aligned} \quad (5)$$

The expected return of the platform {Not Iterate} strategy is obtained in the following equation:

$$U_{22} = \alpha\gamma(R_3 - \tau_2 - L_2) + \alpha(1-\gamma)R_3 + (1-\alpha)\gamma(R_3 - \tau_2 + L_2) + (1-\alpha)(1-\gamma)R_3 \quad (6)$$

The expected return to the platform is obtained from Equations (5) and (6).

$$U_2 = \beta U_{21} + (1-\beta)U_{22} \quad (7)$$

The platform's Dynamic Replication Equation is as follows:

$$\begin{aligned}\phi(\beta) &= \tfrac{d\beta}{dt} = \beta(1-\beta)(U_{21}-U_{22}) \\ &= \beta(1-\beta)\{\alpha[(1-\theta)\Delta C + C_2] + \gamma[(1-e)R + \tau_2 - L_2 - \tau_1] + \tau_1 - (1-\theta)C_0\}\end{aligned} \tag{8}$$

The expected return of the user {Feedback} strategy is obtained in the following equation:

$$\begin{aligned}U_{31} = &\ \alpha\{\beta[R_2 + eR - \theta(C_0 - \Delta C)] + (1-\beta)[R_2 - \theta(C_0 - \Delta C) + \tau_2]\} \\ &+ (1-\alpha)\{\beta(R_2 + eR - \theta C_0) + (1-\beta)(R_2 - \theta C_0 + \tau_2)\}\end{aligned} \tag{9}$$

The expected return of the user {Not Feedback} strategy is obtained in the following equation:

$$U_{32} = \alpha[\beta(R_2 + L_1 - \tau_1) + (1-\beta)R_2] + (1-\alpha)[\beta(R_2 - \tau_1 + L_1) + (1-\beta)R_2] \tag{10}$$

The expected returns of the user under two strategies are as follows:

$$U_3 = \gamma U_{31} + (1-\gamma)U_{32} \tag{11}$$

As a result, the user's Replication Dynamic Equation is obtained by the following calculation:

$$\psi(\gamma) = \frac{d\gamma}{dt} = r(1-\gamma)(U_{31} - U_{32}) = \gamma(1-\gamma)[\alpha\theta\Delta C - \theta C_0 + \beta(eR + \tau_1 - L_1 - \tau_2) + \tau_2)] \tag{12}$$

The replicated dynamic system and its stabilization strategies are obtained through the joint equations of (4), (8) and (12) below.

$$\begin{cases}\varphi(\alpha) = \frac{d\alpha}{dt} \\ \phi(\beta) = \frac{d\beta}{dt} \\ \psi(\gamma) = \frac{d\gamma}{dt}\end{cases} \Longrightarrow \begin{cases}\alpha = 0, \alpha = 1 \\ \beta = 0, \beta = 1 \\ \gamma = 0, \gamma = 1\end{cases} \tag{13}$$

A Jacobian matrix (Equation (14)) is obtained by taking the partial derivatives of Equations (4), (8) and (12).

$$J = \begin{vmatrix}\frac{\partial\varphi(\alpha)}{\partial\alpha} & \frac{\partial\varphi(\alpha)}{\partial\beta} & \frac{\partial\varphi(\alpha)}{\partial\gamma} \\ \frac{\partial\phi(\beta)}{\partial\alpha} & \frac{\partial\phi(\beta)}{\partial\beta} & \frac{\partial\phi(\beta)}{\partial\gamma} \\ \frac{\partial\psi(\gamma)}{\partial\alpha} & \frac{\partial\psi(\gamma)}{\partial\beta} & \frac{\partial\psi(\gamma)}{\partial\gamma}\end{vmatrix} \tag{14}$$

Let Equation (13) equal zero to obtain an equilibrium, and substitute it into Equation (14) to obtain the eigenvalues (Table 5). Identification of the ESS can be determined using the Friedman method, which requires determinants greater than zero $\boldsymbol{det > 0}$ and a trace less than zero $\boldsymbol{tr < 0}$ [92]; or using the Lyapunov method, which requires all eigenvalues of the Jacobian matrix to be less than zero [93]. Based on the eigenvalues in Table 6, we can directly apply the Lyapunov method for ESS identification.

Table 5. Eigenvalues of equilibrium points.

Equilibriums	λ_1	λ_2	λ_3
$E_1(0,0,0)$	$(1-p)R_1 - C_1$	$\tau_1 - (1-\theta)C_0$	$-\theta C_0 + \tau_2$
$E_2(0,1,0)$	$(1-p)R_1 - C_1$	$(1-e)R + \tau_2 - L_2 - (1-\theta)C_0$	$\theta C_0 - \tau_2$
$E_3(0,0,1)$	$(1-p)R_1 - C_1 - C_2$	$-\tau_1 + (1-\theta)C_0$	$-\theta C_0 + eR + \tau_1 - L_1$
$E_4(0,1,1)$	$(1-p)R_1 - C_1 - C_2$	$-(1-e)R - \tau_2 + L_2 + (1-\theta)C_0$	$\theta C_0 - eR - \tau_1 + L_1$

Table 5. *Cont.*

Equilibriums	λ_1	λ_2	λ_3
$E_5(1,0,0)$	$-(1-p)R_1+C_1$	$(1-\theta)\Delta C+C_2+\tau_1-(1-\theta)C_0$	$\theta(\Delta C-C_0)+\tau_2$
$E_6(1,1,0)$	$-(1-p)R_1+C_1$	$(1-\theta)\Delta C+C_2+(1-e)R+\tau_2-L_2-(1-\theta)C_0$	$\theta(C_0-\Delta C)-\tau_2$
$E_7(1,0,1)$	$C_1+C_2-(1-p)R_1$	$-(1-\theta)\Delta C-C_2-\tau_1+(1-\theta)C_0$	$\theta(\Delta C-C_0)+eR+\tau_1-L_1$
$E_8(1,1,1)$	$C_1+C_2-(1-p)R_1$	$-(1-\theta)\Delta C-C_2-(1-e)R-\tau_2+L_2+(1-\theta)C_0$	$\theta(C_0-\Delta C)-eR-\tau_1+L_1$

Table 6. Determination of ESS.

Equilibriums	Scenario1: $\begin{cases} C_2+\tau_1-(1-\theta)(C_0-\Delta C)<0 \\ \textit{and}, \\ \tau_2-\theta(C_0-\Delta C)<0 \end{cases}$				Scenario2: $\begin{cases} \tau_1-(1-\theta)C_0>0, \\ \textit{or}, \\ \tau_2-sC_0>0 \end{cases}$				Scenario3 condition 1: $\begin{cases} C_2+\tau_1-(1-\theta)(C_0-\Delta C)>0, \\ \textit{and}, \\ \tau_1-(1-\theta)C<0 \end{cases}$ or, condition 2: $\begin{cases} \tau_2-s(C_0-\Delta C)>0, \\ \textit{and}, \\ \tau_2-\theta C_0<0 \end{cases}$			
	λ_1	λ_2	λ_3	**Result**	λ_1	λ_2	λ_3	**Result**	λ_1	λ_2	λ_3	**Result**
$E_1(0,0,0)$	*pos*	*neg*	*pos/neg*	**Ns**	*pos*	*pos*	*pos*	**Sd**	*pos*	*neg*	*neg*	**Ns**
$E_2(0,1,0)$	*pos*	*pos*	*neg*	**Ns**	*pos*	*pos/neg*	*neg*	**Ns**	*pos*	*pos*	*pos*	**Sd**
$E_3(0,0,1)$	*pos*	*pos*	*pos*	**Sd**	*pos*	*neg*	*pos*	**Ns**	*pos/neg*	*neg*	*pos*	**Ns**
$E_4(0,1,1)$	*neg*	*neg*	*neg*	**ESS**	*neg*	*pos*	*pos*	**Ns**	*neg*	*pos/neg*	*pos*	**Ns**
$E_5(1,0,0)$	*pos*	*neg*	*neg*	**Ns**	*pos/neg*	*neg*	*neg*	**Ns**	*pos*	*neg*	*neg*	**Ns**
$E_6(1,1,0)$	*neg*	*pos*	*pos*	**Ns**	*neg*	*pos*	*neg*	**Ns**	*neg*	*pos*	*neg*	**Ns**
$E_7(1,0,1)$	*neg*	*pos*	*pos*	**Ns**	*neg*	*pos*	*pos*	**Ns**	*neg*	*neg*	*pos*	**Ns**
$E_8(1,1,1)$	*neg*	*neg*	*neg*	**ESS**	*neg*	*neg*	*neg*	**ESS**	*neg*	*neg*	*neg*	**ESS**

Note for abbreviations: pos—positive; neg—negative; Ns—non-stable point; Sd—saddle point; ESS—Evolutionary Stable Strategy.

The underlying assumptions are as follows: assuming the government incentives have positive effects, when the government chooses to provide incentives and both open-source platforms and users actively accept the incentives, the income of all parties involved is Pareto-improved compared to other decision combinations, which is consistent with limited rationality [94]. This leads to following base constraints:

$$\textit{base constraints}:\left\{\begin{array}{c}(1-p)R_1-C_1-C_2 \\ (1-e)R+\tau_2-L_2-(1-\theta)C_0>0 \\ eR+\tau_1-L_1-\theta C_0\end{array}\right. \tag{15}$$

Through Equation (15), the evolutionary game has following ESS: $E_4(0,1,1)$ and $E_8(1,1,1)$ are two ESSs. $E_4(0,1,1)$ represents the combination of evolutionary stabilization strategies of the G-P-U as {Not Incent $\times$ Iteration $\times$ Feedback}; $E_8(1,1,1)$ represents the combination of evolutionary stabilization strategies of the G-P-U parties as {Incent $\times$ Iteration $\times$ Feedback}. These two ESSs exist under the scenarios below.

$$\textit{Scenario1}:\left\{\begin{array}{c}C_2+\tau_1-(1-\theta)(C_0-\Delta C) \\ \tau_2-\theta(C_0-\Delta C)\end{array}\right.<0 \tag{16}$$

$E_8(1,1,1)$ is still an ESS when the following scenarios are satisfied:

$$\left\{\begin{array}{c}\textit{Scenario2}:\ \tau_1-(1-\theta)C_0>0,\ \textit{or},\tau_2-sC_0>0 \\ \textit{Scenario3}:\ \left\{\begin{array}{c}\textit{condition1}: C_2+\tau_1-(1-\theta)(C_0-\Delta C)>0,\textit{and},\tau_1-(1-\theta)C<0 \\ \textit{condition2}:\ \tau_2-s(C_0-\Delta C)>0,\textit{and},\tau_2-\theta C_0<0\end{array}\right.\end{array}\right. \tag{17}$$

All ESSs are identified by the Lyapunov method in Table 6.

The Theoretical Phase diagram (Figure 3) indicates the evolutionarily stable strategy combination $E_8(1,1,1)$ implies the formation of an open innovation system by open-source platforms and users after government incentives; ESS combination $E_4(0,1,1)$ implies the spontaneous formation of an open innovation system by open-source platforms and users in the absence of government incentives, as well as the formation of an open innovation system with the help of government incentives that is sustained even after withdrawal of such incentives.

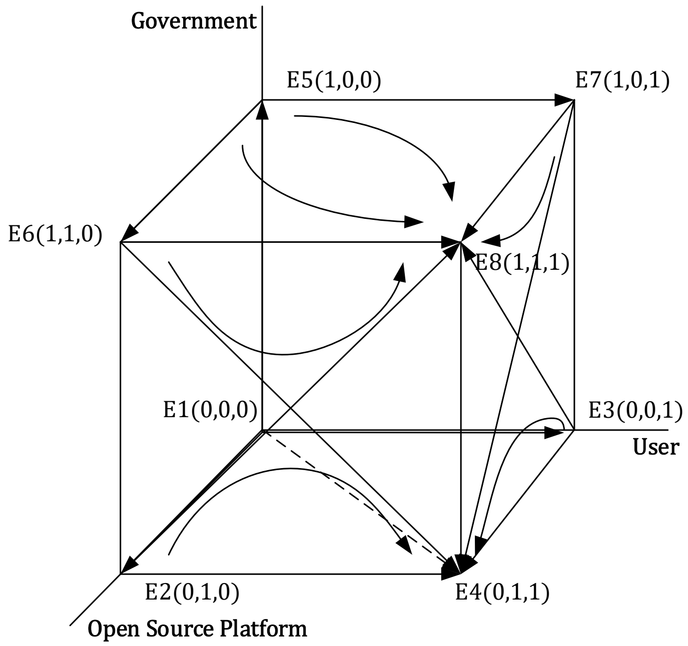


Figure 3. Theoretical Phase diagram.

3.3. Results Analysis

In this subsection, the numerical simulations are implemented. As stated above in Tables 2 and 6, the domain of definition of the above probabilistic, proportionate and economic parameters are within the [0, 1] interval, in line with probability theory or normalized and scaled, and the ESSs are determined based on the Lyapunov method [93].

Therefore, the use of Algorithm 1 below to perform random searches for parameter values can ensure the consistency of the simulation experiments and studies under the above constraints and scenarios.

Algorithm 1 Random Search for Parameter Values

1. Initialize: {parameter_i} ← random numeric sample ∈ [0, 1]
2. Define: {base constraints} in accord. with Equation (15)
3. Define: {scenario constraints} in accord. with Equations (16) and (17)
4. WHILE {base constraints} is False or {scenario constraints} is False:
5. Repeat search on random numeric sample generated
6. Until {base constraints} is True and {scenario constraints} is True
6. End WHILE
7. Output {parameter_i}
8. End Algorithm 1

A set of random samples (floating-point precision results obtained by the algorithmic program rounded to four decimal places) for parameters satisfying both base constraints and Scenario1 are as follows:

$R_1 \approx 0.9838,\ C_1 \approx 0.6952,\ C_2 \approx 0.0629, \Delta C \approx 0.2322,$
$p \approx 0.0268,\ C_0 \approx 0.7403,\ \theta \approx 0.6315,\ R \approx 0.9948,$
$e \approx 0.5832,\ L_1 \approx 0.0311,\ L_2 \approx 0.1708,\ \tau_1 \approx 0.0123,\ \tau_2 \approx 0.2074$

Similarly, a set of random samples for parameters satisfying both base constraints and Scenario2 are as follows:

$R_1 \approx 0.9559,\ C_1 \approx 0.4695,\ C_2 \approx 0.3267,\ \Delta_C \approx 0.2025,$
$p \approx 0.0489,\ C_0 \approx 0.9130,\ \theta \approx 0.36320,\ R \approx 0.4717,$

$e \approx 0.4797,\ L_1 \approx 0.5599,\ L_2 \approx 0.1398,\ \tau_1 \approx 0.7416,\ \tau_2 \approx 0.8507$

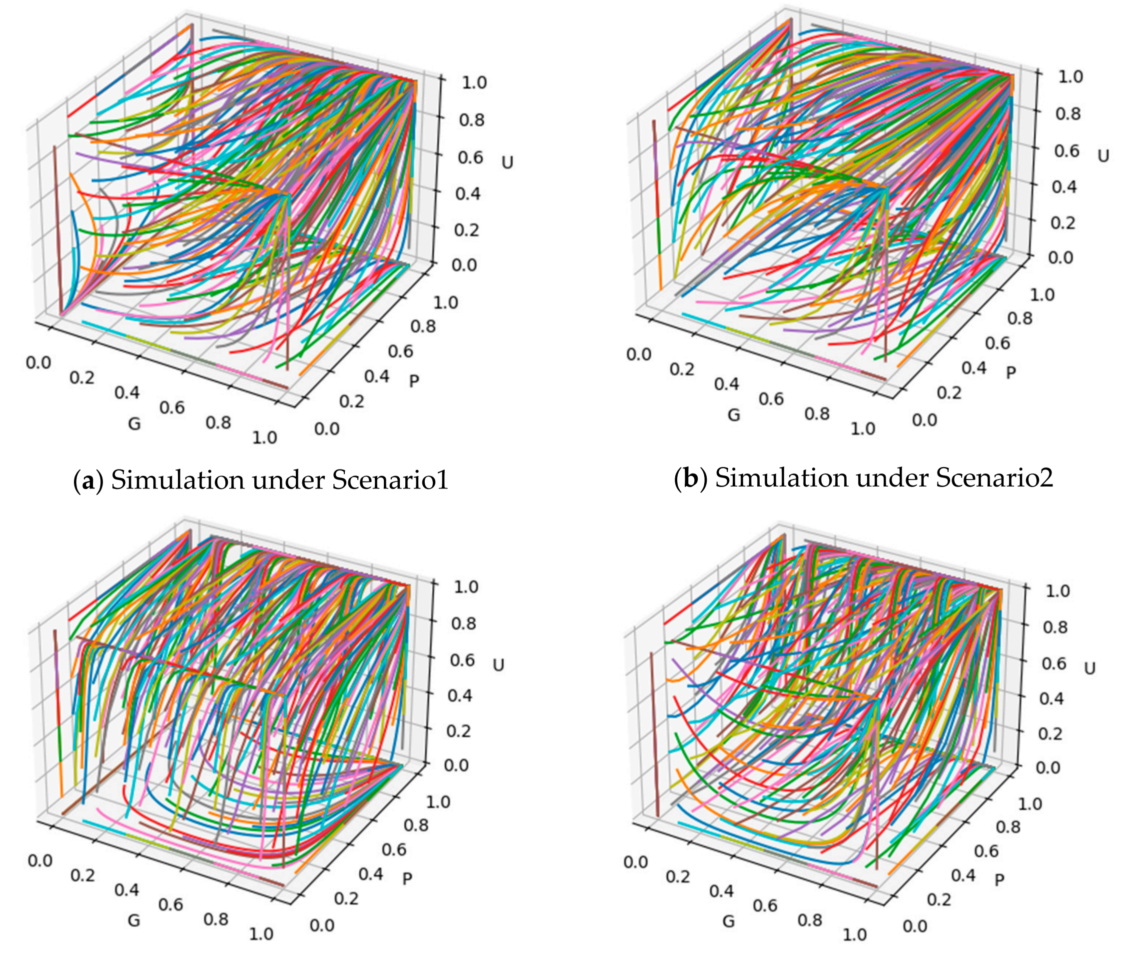
Likewise, two sets of random samples for parameters satisfying both base constraints and condition1 in Scenario3 are as follows:

$R_1 \approx 0.3529,\ C_1 \approx 0.0085,\ C_2 \approx 0.0754,\ \Delta C \approx 0.1823,$
$p \approx 0.6572,\ C_0 \approx 0.2280,\ \theta \approx 0.3560,\ R \approx 0.8437,$
$e \approx 0.4070,\ L_1 \approx 0.0130,\ L_2 \approx 0.4291,\ \tau_1 \approx 0.0439,\ \tau_2 \approx 0.8934$

Additionally, condition2 in Scenario3 are as follows:

$R_1 \approx 0.4486,\ C_1 \approx 0.1851,\ C_2 \approx 0.1083,\ \Delta C \approx 0.6559,$
$p \approx 0.2011,\ C_0 \approx 0.3702,\ \theta \approx 0.7539,\ R \approx 0.4617,$
$e \approx 0.4650,\ L_1 \approx 0.0876,\ L_2 \approx 0.0275,\ \tau_1 \approx 0.8159,\ \tau_2 \approx 0.0343$

Four three-dimensional evolutionary tracking diagrams (Figure 4a–d) were simulated based on the above sets of parameters. In order to analyse the evolutionary formation of the ESS more clearly, we observe the effect of the change in the initial value of the innings on the evolutionary equilibrium in the two-dimensional evolutionary perspective.



(**a**) Simulation under Scenario1

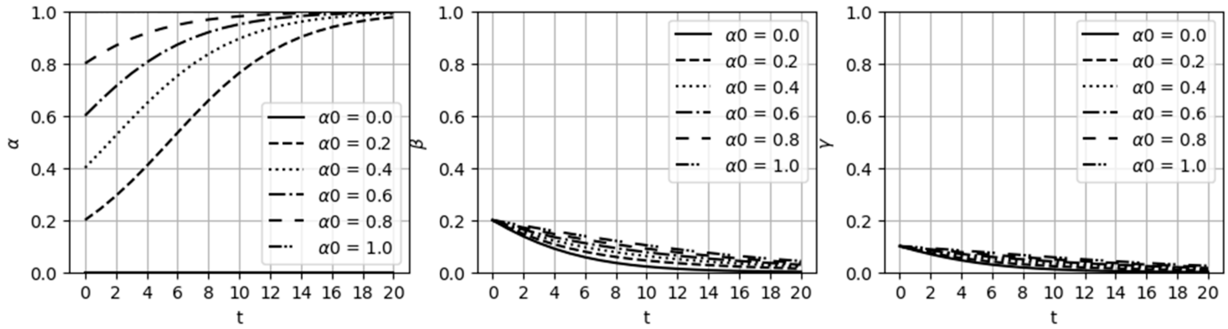
(**b**) Simulation under Scenario2

(**c**) Simulation under condition1 Scenario3

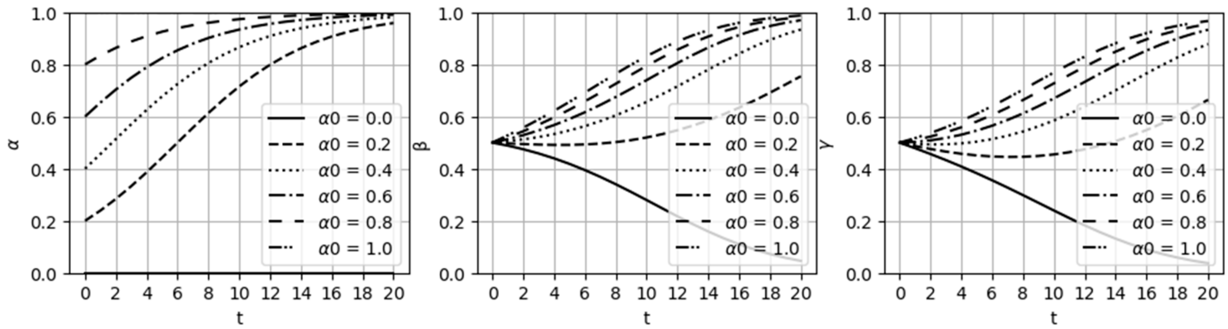
(**d**) Simulation under condition2 Scenario3

Figure 4. Simulations for the three-dimensional evolutionary tracking diagrams. Note: Axis labels G, P, and U in sub-diagrams (**a**–**d**) above represent government, open-source platform, and user, respectively. Color lines are set up for easy visual identification of tracks.

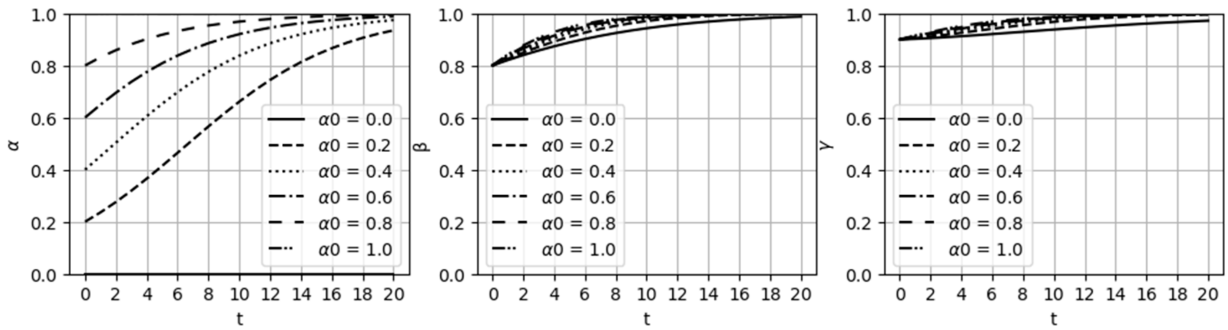
With the underlying assumptions, we note that the probability of players choosing strategies each corresponds to their respective levels of willingness [92,93]. Hence, the impact of their initial willingness value of the selection of government incentive probability on the evolution is further quantitatively simulated by controlling for the probability of open-source platform-enforced diffusion and user-implemented feedback, represented by low to medium and high willingness levels. (Figure 5a–c).



(**a**) Low initial willingness of open-source platforms ($\beta = 0.2$) and users ($\gamma = 0.1$)



(**b**) medium initial willingness of open-source platforms ($\beta = 0.5$) and users ($\gamma = 0.5$)



(**c**) high initial willingness of open-source platforms ($\beta = 0.8$) and users ($\gamma = 0.9$)

Figure 5. Sensitivity impact of the initial willingness selection on evolution.

Figure 5a shows that changes in the initial willingness level of government incentives are difficult to motivate platforms and users to evolve towards open-source diffusion when they are at low willingness levels, corresponding to the unstable $\boldsymbol{ESS_5}(1,0,0)$. Figure 5b shows that at medium willingness levels of both platforms and users, non-zero levels of initial incentives have the opportunity to motivate both platforms and users to evolve towards open-source diffusion, corresponding to the stable $\boldsymbol{ESS_8}(1,1,1)$. Figure 5c further shows that at high levels of willingness, both platforms and users are still able to spontaneously evolve towards open-source diffusion even if the government's initial willingness to incentivize is zero (probability of the government incentive strategy α0 = 0), corresponding to the stable $\boldsymbol{ESS_4}(0,1,1)$.

3.4. Model Tradeoffs and Drawbacks

Open-source diffusion has high complexity [95], involving the ecological niche and intentions of entities in the open-source ecosystem, the technological maturity of software and hardware as objects, as well as the collaborative network formed by collective intelligence. Thus, the model has its tradeoffs and drawbacks.

The model is unable to consider every detailed variable from cultural, institutional, corporate, or user group behaviour due to technological diffusion and institutional factors. Further research in different geographical and institutional contexts is needed to enhance our understanding of open-source diffusion and its impact on enterprise strategy, R&D performance, top-level design, and institutional advantages.

Moreover, while the evolutionary game model provides valuable insights, it makes various assumptions that may filter out real-world randomness and perturbation. Participant decisions are assumed to be rationally limited [88]; however, in reality, they can be influenced by more complex factors like individual preferences and social influence.

In addition, the model conducts a sensitivity analysis to ensure the robustness and reliability of the results under controlled variables. However, the provided ESSs and diffusion path cannot cover all possible variations. In practical applications, it is recommended to combine the results with other methods and models to provide comprehensive and accurate decision support considering other factors such as economics, competition patterns, regulations, and policies.

4. Discussion

4.1. Incentive Step-In and Step-Out: From ESS_8 to ESS_4

In early stages of the diffusion of open-source operating system software, government intervention is a necessary means to propel its development. As an open-source software ecosystem has not yet been established, manufacturers and users have limited awareness and understanding of open-source software. Government intervention could help open-source software gain a market share and improve its development speed. The government could adopt various ways to encourage the development of open-source software such as providing policy support, rewards or subsidies to enterprises adopting open-source software. At this time, incentives can play a leading role. As the open-source software ecosystem gradually improves and the technology of open platform accumulates, user retention will gradually increase, which promotes stable development. During this process, open-source platforms can begin to use its sustained operating income for innovation iteration and community maintenance, thereby further enhancing software value and user experience. Government incentives for the software market gradually become weaker and withdrew from its leading position. As the ecosystem further matures, incentives gradually lose strategic value. Open-source software becomes relatively mature and stable industry, reducing government intervention and interference. The government begins to gradually withdraw from its leading position, and open-source software moves towards a more free-market track. At this point, open-source software has sufficient competitiveness and market share to autonomously participate in market competition and industrial development.

4.2. Accelerating Strategy Implementation through Incentive Platforms

The government often tends to incentivize open-source platforms more than users, in that open-source platforms, as the main subject of technological innovation, gather various economic activities and entities in the market, thus making it easier for governments to exert incentives and regulatory measures. Therefore, governments can more easily achieve their strategic goals for innovation and industrial development. For example, (1) in strategic layout goals of scientific and technological innovation and digital transformation, development of open-source platform enterprises is a key part for governments. By using open-source platforms as a point of entry, governments encourage all kinds of enterprises to use open-source platforms, promote digital transformation and enhance enterprise efficiency and innovation capabilities. (2) In promoting innovation goals, government incentives for open-source platforms help encourage companies to facilitate innovation and R&D on these platforms, improving product quality and performance, owning independent technology, intellectual property rights, and core competitiveness. (3) To drive industrial development goals, government incentives for open-source platform development can promote the development of the entire open-source ecosystem as well as related industries, markets, and ecology, thus promoting the coordinated development of industry and improving market competitiveness. Thus, from a strategic and future development perspective, governments are more willing to incentivize open-source platform development enterprises than users. This strategy of governments can also help improve the overall development level of the industry and domestic digital construction process.

4.3. Increase Willingness to Participate by Incentivizing Users

The willingness of users to participate is a gradual process as follows:

Know and Recognize → Participate and Try → Learn and Master → Contribute and Feedback → Develop and Innovate.

During the Know and Recognize stage, government incentives and promotions expand the market influence of open-source platforms and operating system software, increasing our understanding and recognition of their advantages and characteristics. Government incentives and promotions are mainly carried out through official websites, promotional materials, and promotional activities, which widely publicize information on the characteristics, usage methods, and advantages of open-source platforms and operating system software to the public. These promotional and advertising activities aim to increase users' awareness and understanding of open-source platforms and operating system software, and they make more people aware of their potential value and advantages. At the same time, government support and promotion also established a reputation for open-source platforms and operating system software among users and the market, making them competitive with other commercial software and even gaining more market share in some areas. Government incentives and promotions are an important means of promoting open-source platforms and operating system software, which can provide people with the initial knowledge and understanding of open-source platforms and lay a foundation for their subsequent use and involvement.

In the Participate and Try stage, after having a certain level of understanding of open-source platforms, some users will start to use and participate in the open-source project, contributing code, providing feedback and suggestions, and other ways to understand the open-source platform, thus improving their familiarity and trust in the open-source platform.

During the Learn and Master stage, after a period of participation and trial, some users may develop a strong interest in the open-source platform and decide to delve deeper into the relevant technologies and knowledge, thus investing more time and effort.

During the Contribute and Feedback stage, as some users become familiar with and master the relevant knowledge and technologies, they may begin to contribute code, submit bug reports and suggestions, and assist in improving and refining the open-source platform.

During the Develop and Innovate stage, some users may adopt an open-source platform as their primary technology and development platform, and actively participate in other open-source projects, gradually integrating into the developer and open source community.

As can be seen, with government encouragement and support, the willingness of these users to participate in open-source platforms is gradually increasing, and more people are willing to join in and contribute to their development and improvement. In the process of continuous innovation, iteration, and improvement of the ecosystem and reputation of open-source platform, it continues to attract more users to try it out and gradually develop user retention. The government also helps open-source platforms understand users' needs and pain points, continuously improving and optimizing products and services by encouraging users to use open-source operating system software and provide feedback. This mutual promotion and collaboration strengthen user participation and value perception, continuously driving the development and growth of open-source platforms and operating system software.

4.4. Mitigating Supply Chain Risks

The diffusion of open-source operating system software is a means of reducing risk of software supply disruption. Open source implies that the source code is made public and can be freely copied, distributed, and modified. Compared with proprietary software, open-source software is more transparent and open, allowing maintenance to be shifted to the community and developers, thereby reducing the dependence on software vendors. Therefore, with open-source operating system software as a foundation, the power of community and developers can be fully utilized to expand the scope and depth of updates and maintenance, thus reducing the risk of software supply disruption caused by supplier failure, bankruptcy, etc. Open-source operating system software can promote software interoperability and compatibility. With an open source code, developers can modify and customize software to better adapt to different hardware and software environments. This flexible feature can promote software interoperability and compatibility, avoid control of software ecosystems by a single vendor, and reduce the risk of software supply disruption. Therefore, diffusion of open-source operating system software is a prerequisite for reducing the risk of supply disruption, and active participation and cooperation of the government, open-source platforms, and users can promote the development and popularization of open-source software.

4.5. Implications

We propose the following recommendations to enhance the diffusion of open-source operating system software and address the risk of supply chain disruption.

(1) Multi-channel procurement. When procuring software, enterprise users should consider multiple channels to reduce their reliance on a single supplier and ensure the ability to switch to other suppliers at any time. For example, when purchasing operating system software, using products from multiple vendors should be considered, such as using open-source operating systems for some servers and commercial operating systems for others.
(2) Active participation in open source communities. Enterprises should actively participate in open-source platforms, trust and support software development from open-source platforms, and participate in product research and testing. By participating in open source communities, enterprises can have a better understanding of the software development and maintenance process, and make better decisions in the event of supply chain disruptions.
(3) Independent development. Enterprises should explore independent development and use open-source operating system software source code for secondary development and customization. This can avoid the reliance on suppliers, enable control over the software development and maintenance process, and reduce the risk of supply chain disruptions.
(4) Enhance and adopt virtualization and containerization technologies. Decoupling and isolating applications and operating system software using virtual machines and containerization technologies can make it easier to migrate and manage applications and operating system software, thereby reducing the risk of supply chain disruption.
(5) Actively adopt backup and recovery strategies. Enterprises should regularly backup and archive data and software to enable timely recovery in the event of a supply chain disruption. Enterprises should develop a comprehensive emergency plan, including backup strategies, backup recovery testing, and disaster recovery procedures, among others.

In summary, in order to reduce the risk of supply chain disruption, enterprises can transition to the diffusion of open-source operating system software by adopting multiple procurement channels, actively participating in open source communities, conducting independent research and development, utilizing virtualization and containerization technologies, and implementing backup and recovery strategies, among other approaches. Essentially, the recommendations to mitigate the risk of supply chain disruption involve active transformation of enterprise users into developer roles in open-source platforms. This strategy benefits enterprises by reducing the risk of supply chain disruption and promoting their own development. As supporters and contributors of open source, enterprises can participate in development and testing in open source communities to help improve and expand open-source software to meet their own needs. This approach helps to increase enterprise understanding and mastery of open-source software, reduce reliance on specific suppliers, establish their own technological advantages and competitiveness, and promote the development of the entire open source ecosystem. Additionally, active participation in open source communities can bring additional benefits to enterprises, such as being recognized as industry leaders, accelerating development speed, improve problem solving, and building a better corporate reputation, among others. Therefore, transforming from enterprise users to contributors in open source communities is not only a defensive measure to reduce the risk of supply chain disruption, but also an important avenue for technological innovation, competitive advantage, and corporate reputation.

5. Conclusions

Android–iOS–Windows has a first-mover advantage and mature market, with threats of commercial monopoly and technological supply cut-off. However, it also poses threats of commercial monopolies and technology disruptions. Adopting open-source paradigms for independent innovation and technological independence is a potential solution to these threats. Open-source diffusion is closely related to the innovation iteration of open-source platforms and the increasing size of the user base. Specifically, based on an open collaborative model, open source continuously attracts more users, contributors, and developers to improve and perfect it with collective intelligence. Its excellent performance and code quality often attract more users to join, further enhancing the reputation and influence of open-source software. As the application scenarios of open source become more extensive and deeper, its user base will continue to grow, attracting more developers and contributors to join the open source community, jointly promoting the innovation and iteration of open-source software and forming a virtuous cycle. Therefore, expansion of open source and growth of its user base are important drivers for innovation iteration in smart digital infrastructure and key factors in maintaining continuous development in the industrial ecosystem.

This study analyses open-source diffusion among the government, open-source platforms, and users as stakeholders using an evolutionary game framework. The study found two evolutionarily stable strategy combinations as follows: {Not Incent × Iteration × Feedback} and {Incent × Iteration × Feedback}. This shows that open-source platforms and users generate open-source diffusion and form an open innovation system with government incentives; or they spontaneously create an open innovation system with open-source diffusion in the absence of government incentives as well as generate open-source diffusion initially with the help of government incentives and still maintain an open innovation system after the incentives have ceased.

Further analysis indicates that the government is more inclined to incentivize open-source platforms because they can gather open-source innovation entities and various economic activities, making them more conducive to incentives and regulation. Government incentives can help achieve strategic and targeted development goals for open source. Government involvement in incentives can prompt open-source platforms and users to form an open innovation system, establish and improve the open-source operating system software ecosystem, and accelerate open-source diffusion. As open-source diffusion scales up, the software industry's ability for independent innovation based on open source increases, enhancing its competitiveness against proprietary software, gradually reducing the government's dominant role. Under the incentives, user retention increases, actively participating in open source feedback, joining open source innovation collaborations, and promoting open-source diffusion.

In conclusion, open-source diffusion is an important way to mitigate software supply chain disruption and technology supply disruption. Active participation and cooperation among the government, open-source platforms, and users can promote the development and popularization of open-source software. We suggest further strengthening of open-source diffusion to cope with supply disruption risks, such as multi-channel software procurement, encouraging participation in open source innovation, incentivizing open-source platform companies to conduct independent research and development, enhancing and applying virtualization and containerization technologies, and implementing flexible and robust disaster recovery backup strategies.

This study contributes by considering the stakeholders involved in open-source diffusion and using an evolutionary game model for analysis. Results provide theoretical value and a practical reference for the development of the open-source software engineering industry. Additionally, our study is strategically significant for sustainable technological development, especially in the context of digital transformation and data-driven economies.

The main limitation of this study is the lack of research on the software ecology of different open-source operating systems, including various kernels and distributions. Therefore, in future research, case studies on the innovation and diffusion of typical distributions of open-source operating systems, such as Linux kernel, Debian, Android, Chrome OS, etc., must be conducted using a cross-disciplinary approach towards software engineering and management science.

Author Contributions: Conceptualization, T.L. and J.Z.; Methodology, T.L.; Software, T.L.; Validation, T.L., J.Z. and J.L.; Formal analysis, T.L.; Investigation, T.L.; Resources, T.L.; Data curation, T.L.; Writing—original draft, T.L.; Writing—review & editing, T.L., J.Z., J.L., C.Y. and B.Z.; Visualization, T.L.; Supervision, J.Z., J.L. and C.Y.; Project administration, J.Z. and B.Z.; Funding acquisition, B.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was been supported by (1) the Research Project of National Academy of Innovation Strategy (NAIS) "Research on the application of key technologies in digital economy at home and abroad" (2022-sjzx-02); (2) National Social Science Fund (NSSFC) project "Research on the Modernization of Intellectual Property Governance for Digital Innovation" (22VRC064); (3) National Social Science Fund Key Project (NSSFKP) "Research on Goals, Major Directions, and Strategies for Winning the Battle of Key Core Technology under the New Comprehensive National System" (23AZD038).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing is not applicable to this article as no new data were created or analysed in this study.

Acknowledgments: The authors express their thanks to the Jiangsu Decision-Making and Consultation Institution of Service-Oriented Government Development fund from Nanjing University of Science and Technology.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Nogueira, E.; Gomes, S.; Lopes, J.M. Triple Bottom Line, Sustainability, and Economic Development: What Binds Them Together? A Bibliometric Approach. *Sustainability* **2023**, *15*, 6706. [CrossRef]
2. Mcgahan, A.M.; Bogers, M.L.A.M.; Chesbrough, H.; Holgersson, M. Tackling Societal Challenges with Open Innovation. *Calif. Manag. Rev.* **2021**, *63*, 49–61. [CrossRef]
3. Perens, B. The emerging economic paradigm of Open Source. *First Monday* **2005**, *10*. [CrossRef]

4. Yigitcanlar, T.; Mehmood, R.; Corchado, J.M. Green Artificial Intelligence: Towards an Efficient, Sustainable and Equitable Technology for Smart Cities and Futures. *Sustainability* **2021**, *13*, 8952. [CrossRef]
5. Alotaibi, S.; Mehmood, R.; Katib, I.; Rana, O.; Albeshri, A. Sehaa: A Big Data Analytics Tool for Healthcare Symptoms and Diseases Detection Using Twitter, Apache Spark, and Machine Learning. *Appl. Sci.* **2020**, *10*, 1398. [CrossRef]
6. Bonvoisin, J. Implications of Open Source Design for Sustainability. In *Proceedings of the International Conference on Sustainable Design and Manufacturing, Cham, Switzerland, 1 January 2016*; Setchi, R., Howlett, R.J., Liu, Y., Theobald, P., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 49–59.
7. Dong, X.; Jin, S.; Lu, X. Comparison of Xiaomi Smart Home and Huawei HarmonyOS Business Ecosystem. In *Proceedings of the 2022 International Conference on Business and Policy Studies, Singapore, 1 January 2022*; Li, X., Yuan, C., Ganchev, I., Eds.; Springer Nature: Singapore, 2022; pp. 640–648.
8. Kalok, S. The Analysis of Smartphones' Operating System and Customers' Purchasing Decision: Application to HarmonyOS and Other Smartphone Companies. In Proceedings of the 2022 7th International Conference on Financial Innovation and Economic Development (ICFIED 2022), Harbin, China, 1 January 2022; Atlantis Press: Dordrecht, The Netherlands, 2022; pp. 417–421.
9. Oh, S.; Seo, B.; Hahm, C.; Song, J.; Lee, J.; Lee, T. Dynamic boosting for fast user interactivity in smart devices using Tizen OS. In Proceedings of the 2016 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia), Seoul, Republic of Korea, 26–28 October 2016; pp. 1–3.
10. Liu, S.; Rong, X.; Li, S.; Dong, A. Design of harmony OS-based IOT system for wearable health monitoring. In Proceedings of the International Conference on Internet of Things and Machine Learning (IoTML 2021), Dalian, China, 17–19 December 2021.
11. Cui, P.; Liu, Z.; Bai, J. Linux Storage I/O Performance Optimization Based on Machine Learning. In Proceedings of the 2022 4th International Conference on Natural Language Processing (ICNLP), Xi'an, China, 25–27 March 2022; pp. 552–557.
12. Ugur, M.; Jiang, C.; Erf, A.; Ahmed Khan, T.; Kasikci, B. One Profile Fits All. *Acm Sigops Oper. Syst. Rev.* **2022**, *56*, 26–33. [CrossRef]
13. Haim Faridian, P.; Neubaum, D.O. Ambidexterity in the age of asset sharing: Development of dynamic capabilities in open source ecosystems. *Technovation* **2021**, *99*, 102125. [CrossRef]
14. Yu, L.; Schach, S.R.; Chen, K.; Heller, G.Z.; Offutt, J. Maintainability of the kernels of open-source operating systems: A comparison of Linux with FreeBSD, NetBSD, and OpenBSD. *J. Syst. Softw.* **2006**, *79*, 807–815. [CrossRef]
15. Lund, A.; Zukerfeld, M. (Eds.) Profiting from Free and Open Source Software. In *Corporate Capitalism's Use of Openness: Profit for Free?* Springer International Publishing: Cham, Switzerland, 2020; pp. 109–147.
16. Välimäki, M.; Oksanen, V. The impact of free and open source licensing on operating system software markets. *Telemat. Inform.* **2005**, *22*, 97–110. [CrossRef]
17. Sokol, D.D.; Zhu, F. Harming Competition and Consumers under the Guise of Protecting Privacy: An Analysis of Apple's iOS 14 Policy Updates Essay. *Cornell Law Rev. Online* **2021**, *107*, 94. [CrossRef]
18. Jenkins, G.T.; Bing, R.W. Microsofts Monopoly: Anti-Competitive Behavior, Predatory Tactics, and the Failure of Governmental Will. *J. Bus. Econ. Res.* **2007**, *5*, 1–16. [CrossRef]
19. Schmidt, K.M.; Schnitzer, M. Public subsidies for open source? Some economic policy issues of the software market. *Harv. J. Law Technol.* **2003**, *16*, 473. [CrossRef]
20. Willinsky, J. The unacknowledged convergence of open source, open access, and open science. *First Monday* **2005**, *10*. [CrossRef]
21. Nørskov, S.V.; Rask, M. Observation of Online Communities: A Discussion of Online and Offline Observer Roles in Studying Development, Cooperation and Coordination in an Open Source Software Environment. *Forum Qual. Sozialforschung/Forum Qual. Soc. Res.* **2011**, *12*, 5.
22. Schweik, C.; Evans, T.; Grove, J.M. Open source and open content: A framework for global collaboration in social-ecological research. *Ecol. Soc.* **2005**, *10*, 33. Available online: http://www.ecologyandsociety.org/vol10/iss1/art33/ (accessed on 29 September 2023). [CrossRef]
23. Lerner, J.; Tirole, J. Some Simple Economics of Open Source. *J. Ind. Econ.* **2002**, *50*, 197–234. [CrossRef]
24. Lerner, J.; Tirole, J. The Economics of Technology Sharing: Open Source and Beyond. *J. Econ. Perspect.* **2005**, *19*, 99–120. [CrossRef]
25. Shin, D. Ubiquitous city: Urban technologies, urban infrastructure and urban informatics. *J. Inf. Sci.* **2009**, *35*, 515–526. [CrossRef]
26. Yigitcanlar, T.; Kamruzzaman, M. Smart Cities and Mobility: Does the Smartness of Australian Cities Lead to Sustainable Commuting Patterns? *J. Urban Technol.* **2019**, *26*, 21–46. [CrossRef]
27. Batty, M. Artificial intelligence and smart cities. *Environ. Plan. B Urban Anal. City Sci.* **2018**, *45*, 3–6. [CrossRef]
28. Zhang, Y.; Geng, P.; Sivaparthipan, C.B.; Muthu, B.A. Big data and artificial intelligence based early risk warning system of fire hazard for smart cities. *Sustain. Energy Technol.* **2021**, *45*, 100986. [CrossRef]
29. Ullah, Z.; Al-Turjman, F.; Mostarda, L.; Gagliardi, R. Applications of Artificial Intelligence and Machine learning in smart cities. *Comput. Commun.* **2020**, *154*, 313–323. [CrossRef]
30. Yigitcanlar, T.; Desouza, K.; Butler, L.; Roozkhosh, F. Contributions and Risks of Artificial Intelligence (AI) in Building Smarter Cities: Insights from a Systematic Review of the Literature. *Energies* **2020**, *13*, 1473. [CrossRef]
31. Wang, S.; Cao, J. AI and Deep Learning for Urban Computing. In *Urban Informatics*; Shi, W., Goodchild, M.F., Batty, M., Kwan, M., Zhang, A., Eds.; Springer Singapore: Singapore, 2021; pp. 815–844.
32. Etro, F.; Caffarra, C. On the economics of the Android case. *Eur. Compet. J.* **2017**, *13*, 282–313. [CrossRef]

33. Mateos-Garcia, J.; Steinmueller, W.E. The institutions of open source software: Examining the Debian community. *Inf. Econ. Policy* **2008**, *20*, 333–344. [CrossRef]
34. Tabassum, M.; Mathew, K. Software evolution analysis of linux (Ubuntu) OS. In Proceedings of the 2014 International Conference on Computational Science and Technology (ICCST), Kota Kinabalu, Malaysia, 1 January 2014; pp. 1–7.
35. Imambi, S.; Prakash, K.B.; Kanagachidambaresan, G.R. PyTorch. In *Programming with TensorFlow: Solution for Edge Computing Applications*; Prakash, K.B., Kanagachidambaresan, G.R., Eds.; Springer International Publishing: Cham, Switzerland, 2021; pp. 87–104.
36. Virtanen, P.; Gommers, R.; Oliphant, T.E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; et al. SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nat. Methods* **2020**, *17*, 261–272. [CrossRef] [PubMed]
37. Economides, N.; Katsamakas, E. Two-Sided Competition of Proprietary vs. Open Source Technology Platforms and the Implications for the Software Industry. *Manag. Sci.* **2006**, *52*, 1057–1071. [CrossRef]
38. Powell, A.B. Open culture and innovation: Integrating knowledge across boundaries. *Media Cult. Soc.* **2015**, *37*, 376–393. [CrossRef]
39. Yun, J.J.; Zhao, X.; Jung, K.; Yigitcanlar, T. The Culture for Open Innovation Dynamics. *Sustainability* **2020**, *12*, 5076. [CrossRef]
40. Yigitcanlar, T.; Kankanamge, N.; Regona, M.; Maldonado, A.R.; Rowan, B.; Ryu, A.; Desouza, K.C.; Corchado, J.M.; Mehmood, R.; Li, R.Y.M. Artificial Intelligence Technologies and Related Urban Planning and Development Concepts: How Are They Perceived and Utilized in Australia? *J. Open Innov. Technol. Mark. Complex.* **2020**, *6*, 187. [CrossRef]
41. Chesbrough, H. *Open Innovation: The New Imperative for Creating and Profiting from Technology*; Harvard Business School Press: Boston, MA, USA, 2003; p. 227.
42. Didenko, N.; Skripnuk, D.; Kikkas, K.; Kalinina, O.; Kosinski, E. The Impact of Digital Transformation on the Micrologistic System, and the Open Innovation in Logistics. *J. Open Innov. Technol. Mark. Complex.* **2021**, *7*, 115. [CrossRef]
43. Pedersen, K. What can open innovation be used for and how does it create value? *Gov. Inform. Q.* **2020**, *37*, 101459. [CrossRef]
44. Chesbrough, H. The Logic of Open Innovation: Managing Intellectual Property. *Calif. Manag. Rev.* **2003**, *45*, 33–58. [CrossRef]
45. Robertsone, G.; Lapiņa, I. Digital transformation as a catalyst for sustainability and open innovation. *J. Open Innov. Technol. Mark. Complex.* **2023**, *9*, 100017. [CrossRef]
46. Christensen, J.F.; Olesen, M.H.; Kjær, J.S. The industrial dynamics of Open Innovation—Evidence from the transformation of consumer electronics. *Res. Policy* **2005**, *34*, 1533–1549. [CrossRef]
47. Bereznoy, A.; Meissner, D.; Scuotto, V. The intertwining of knowledge sharing and creation in the digital platform based ecosystem. A conceptual study on the lens of the open innovation approach. *J. Knowl. Manag.* **2021**, *25*, 2022–2042. [CrossRef]
48. Yun, J.J.; Cooke, P.; Jung, K.; Yang, B. Theme issue: Open innovation and 'catch-up': Globalist or localist? *Eur. Plan. Stud.* **2023**, *31*, 845–861. [CrossRef]
49. Gassmann, O.; Enkel, E.; Chesbrough, H. The future of open innovation. *R&D Manag.* **2010**, *40*, 213–221.
50. Bogers, M.; Chesbrough, H.; Moedas, C. Open innovation: Research, practices, and policies. *Calif. Manag. Rev.* **2018**, *60*, 5–16. [CrossRef]
51. Yun, J.J.; Ahn, H.J.; Lee, D.S.; Park, K.B.; Zhao, X. Inter-rationality; Modeling of bounded rationality in open innovation dynamics. *Technol. Forecast. Soc.* **2022**, *184*, 122015. [CrossRef]
52. Yun, J.; Pyka, A.; Lee, C.; Won, D.; Kim, D.; Jeong, E.S.; Kodama, F.; Schiuma, G.; Park, H.; Jeon, J.; et al. *Open Innovation Dynamics; Capitalism, Socialism, and Democracy in the 21st Century*; Cambridge Scholars Publish: Newcastle upon Tyne, UK, 2023; p. 527.
53. Olson, D.L.; Rosacker, K. Crowdsourcing and open source software participation. *Serv. Bus.* **2013**, *7*, 499–511. [CrossRef]
54. Hansen, M.; Köhntopp, K.; Pfitzmann, A. The Open Source approach—Opportunities and limitations with respect to security and privacy. *Comput Secur.* **2002**, *21*, 461–471. [CrossRef]
55. Witten, B.; Landwehr, C.; Caloyannides, M. Does open source improve system security? *IEEE Softw.* **2001**, *18*, 57–61. [CrossRef]
56. Colombo, M.G.; Piva, E.; Rossi-Lamastra, C. Open innovation and within-industry diversification in small and medium enterprises: The case of open source software firms. *Res. Policy* **2014**, *43*, 891–902. [CrossRef]
57. Alpern, B.; Augart, S.; Blackburn, S.M.; Butrico, M.; Cocchi, A.; Cheng, P.; Dolby, J.; Fink, S.; Grove, D.; Hind, M.; et al. The Jikes Research Virtual Machine project: Building an open-source research community. *IBM Syst. J.* **2005**, *44*, 399–417. [CrossRef]
58. Yun, J.J.; Zhao, X. Business Model Innovation through a Rectangular Compass: From the Perspective of Open Innovation with Mechanism Design. *J. Open Innov. Technol. Mark. Complex.* **2020**, *6*, 131. [CrossRef]
59. Yun, J.J.; Zhao, X.; Park, K.; Della Corte, V.; Del Gaudio, G. The way to the 'comedy of commons' of a new business model-finding from Naples in Italy, and Jeju Island in South Korea. *Eur. Plan. Stud.* **2023**, *31*, 947–973. [CrossRef]
60. Yigitcanlar, T.; Han, H.; Kamruzzaman, M.; Ioppolo, G.; Sabatini-Marques, J. The making of smart cities: Are Songdo, Masdar, Amsterdam, San Francisco and Brisbane the best we could build? *Land Use Policy* **2019**, *88*, 104187. [CrossRef]
61. Yigitcanlar, T.; Cugurullo, F. The Sustainability of Artificial Intelligence: An Urbanistic Viewpoint from the Lens of Smart and Sustainable Cities. *Sustainability* **2020**, *12*, 8548. [CrossRef]
62. Yigitcanlar, T.; Corchado, J.M.; Mehmood, R.; Li, R.Y.M.; Mossberger, K.; Desouza, K. Responsible Urban Innovation with Local Government Artificial Intelligence (AI): A Conceptual Framework and Research Agenda. *J. Open Innov. Technol. Mark. Complex.* **2021**, *7*, 71. [CrossRef]

63. Ljungberg, J. Open source movements as a model for organising. *Eur. J. Inform. Syst.* **2017**, *9*, 208–216. [CrossRef]
64. Jullien, N.; Stol, K.; Herbsleb, J.D. A Preliminary Theory for Open-Source Ecosystem Microeconomics. In *Towards Engineering Free/Libre Open Source Software (FLOSS) Ecosystems for Impact and Sustainability: Communications of NII Shonan Meetings*; Fitzgerald, B., Mockus, A., Zhou, M., Eds.; Springer: Singapore, 2019; pp. 49–68.
65. Zhang, Y.; Zhou, M.; Stol, K.; Wu, J.; Jin, Z. How Do Companies Collaborate in Open Source Ecosystems? An Empirical Study of OpenStack. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering, Seoul, Republic of Korea, 27 June–19 July 2020*; Association for Computing Machinery: New York, NY, USA, 2020; pp. 1196–1208.
66. Axelrod, W. Mitigating Software Supply Chain Risk. *ISACA J.* **2013**, *4*.
67. Khojasteh, Y. *Supply Chain Risk Management*; Springer: Berlin/Heidelberg, Germany, 2018.
68. Aqlan, F. A software application for rapid risk assessment in integrated supply chains. *Expert Syst. Appl.* **2016**, *43*, 109–116. [CrossRef]
69. Sabbagh, B.A.; Kowalski, S. A Socio-technical Framework for Threat Modeling a Software Supply Chain. *IEEE Secur. Priv.* **2015**, *13*, 30–39. [CrossRef]
70. He, J.; Alavifard, F.; Ivanov, D.; Jahani, H. A real-option approach to mitigate disruption risk in the supply chain. *Omega* **2019**, *88*, 133–149. [CrossRef]
71. Lustenberger, P.; Schumacher, F.; Spada, M.; Burgherr, P.; Stojadinovic, B. Assessing the Performance of the European Natural Gas Network for Selected Supply Disruption Scenarios Using Open-Source Information. *Energies* **2019**, *12*, 4685. [CrossRef]
72. Amankwah, R.; Chen, J.; Kudjo, P.K.; Towey, D. An empirical comparison of commercial and open-source web vulnerability scanners. *Softw. Pract. Exp.* **2020**, *50*, 1842–1857. [CrossRef]
73. Erickson, J.; Brydon, M.; Vining, A.R. Adoption, Improvement, and Disruption: Predicting the Impact of Open Source Applications in Enterprise Software Markets. *J. Database Manag.* **2009**, *19*, 22.
74. Tucci, C.L.; Kaufman, A.; Wood, C.H.; Theyel, G. Collaboration and Teaming in the Software Supply Chain. *Supply Chain Forum An. Int. J.* **2005**, *6*, 16–28. [CrossRef]
75. Shekarian, M.; Mellat Parast, M. An Integrative approach to supply chain disruption risk and resilience management: A literature review. *Int. J. Logist. Res. Appl.* **2021**, *24*, 427–455. [CrossRef]
76. Blanc, M.F.H. Open Source Innovation in Physical Products: Advantages and Disadvantages, a Corporate Perspective. Doctoral Dissertation, Queensland University of Technology, Brisbane City, Australia, 2011.
77. Vukovic, V.; Raković, L. Open Source Approach in Software Development—Advantages and Disadvantages. *Int. Sci. J. Manag. Inf. Syst.* **2008**, *3*, 29–33.
78. Schumpeter, J. *The Theory of Economic Development*; Harvard Economic Studies 46; Harvard Business School Press: Boston, MA, USA, 1911.
79. Costa, Á.; Fernandes, R. Urban public transport in Europe: Technology diffusion and market organisation. *Transp. Res. Part A Policy Pract.* **2012**, *46*, 269–284. [CrossRef]
80. Nicoletti, G.; von Rueden, C.; Andrews, D. Digital technology diffusion: A matter of capabilities, incentives or both? *Eur. Econ. Rev.* **2020**, *128*, 103513. [CrossRef]
81. Mahajan, V.; Muller, E.; Bass, F.M. New Product Diffusion Models in Marketing: A Review and Directions for Research. *J. Mark.* **1990**, *54*, 1–26. [CrossRef]
82. Ma, Y.; Fakhoury, S.; Christensen, M.; Arnaoudova, V.; Zogaan, W.; Mirakhorli, M. Automatic Classification of Software Artifacts in Open-Source Applications. In *Proceedings of the 15th International Conference on Mining Software Repositories, Gothenburg, Sweden, 28–29 May 2018*; Association for Computing Machinery: New York, NY, USA, 2018; pp. 414–425.
83. Temizkan, O.; Kumar, R.L. Exploitation and Exploration Networks in Open Source Software Development: An Artifact-Level Analysis. *J. Manag. Inform. Syst.* **2015**, *32*, 116–150. [CrossRef]
84. Hellman, J.; Cheng, J.; Guo, J.L.C. Facilitating Asynchronous Participatory Design of Open Source Software: Bringing End Users into the Loop. In *Proceedings of the Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems, Yokohama, Japan, 8–13 May 2021*; Association for Computing Machinery: New York, NY, USA, 2021.
85. Bonaccorsi, A.; Rossi, C. Why Open Source software can succeed. *Res. Policy* **2003**, *32*, 1243–1258. [CrossRef]
86. Shanker, A. A Customer Value Creation Framework for Businesses That Generate Revenue with Open Source Software. *Technol. Innov. Manag.* **2012**, *2*, 18–22. [CrossRef]
87. Long, Y.; Siau, K. Social network structures in open source software development teams. *J. Database Manag.* **2007**, *18*, 25–40. [CrossRef]
88. Rubinstein, A. Comments on the Interpretation of Game Theory. *Econometrica* **1991**, *59*, 909–924. [CrossRef]
89. Lu, S.; Zhu, G.; Dai, J. Willingness intensity and co-evolution of decision rationality depending on aspiration enhance cooperation in the spatial public goods game. *PLoS ONE* **2023**, *18*, e0280015. [CrossRef]
90. Hurkens, S.; Schlag, K.H. Evolutionary insights on the willingness to communicate. *Int. J. Game Theory* **2003**, *31*, 511–526. [CrossRef]
91. Rasmusen, E. *Games and Information: An Introduction to Game Theory*, 4th ed.; Wiley-Blackwell: Hoboken, NJ, USA, 2006; p. 560.
92. Friedman, D. Evolutionary Games in Economics. *Econometrica* **1991**, *59*, 637–666. [CrossRef]
93. Lyapunov, A.M. The general problem of the stability of motion. *Int. J. Control* **1992**, *55*, 531–534. [CrossRef]

94. Rabin, M. Incorporating Limited Rationality into Economics. *J. Econ. Lit.* **2013**, *51*, 528–543. [CrossRef]
95. Bryant, A. Liquid uncertainty, chaos and complexity: The gig economy and the open source movement. *Thesis Elev.* **2020**, *156*, 45–66. [CrossRef]