A TECH-NEUTRAL MECHANISM FOR AVOIDING CLOUDBURST DDoS IN REAL-TIME TRANSACTION CLOUD SYSTEM

Tao LI[1,2], Zemin ZHANG[3], Yingjun DU[4]*

Incidents at Shanghai Stock Exchange and Alipay in 2024 highlighted cloudburst challenges in real-time trading. Contemporary systems rely on RTCS, prioritizing trade data consistency over high availability, resulting in cloudburst DDoS due to locked resources. Capacity expansion mitigates request surges but not lock-in issues. Killing locks resolves resource release but may cause unjust outcomes, akin to the 'Trolley Dilemma'. We analyze cloudburst causes and technological injustices of scaling and killing locks. A proposed tech-neutral mechanism uses cognitive AI to manage high-frequency requests, analyzed via evolutionary game theory, considering costs, benefits, and information dynamics in trading and cloud services.

Keywords: Cloudburst; Real-time Transaction Cloud-computing System; Tech-Neutral Mechanism Design; Big Data Platform

1. Introduction

The development of 5G+ infrastructure has released the transmission bottleneck for cloud computing [1], however, it also brings the risk of cloudbursts [2]. 2024 SHSE (Shanghai Securities Exchange) as well as Alipay downtime events exposed the technical weaknesses of RTCS (Real-time Transaction Cloud-computing System) against cloudbursts, which is a reminder for researchers and technologists. Modern systems for securities trading commonly utilize cloud architecture [3]. With high-frequency instructions from automated quantitative techniques, OB(order book) price-first and time-first match-making rules for ensuring service scaling and trading account consistency lead to resource lock-in competition at the expense of high availability [4], making them more vulnerable

[1] Ph.D., Lecturer, School of Artificial Intelligence Technology, Guangxi Technological College of Machinery and Electricity, Nanning, China; e-mail: litao1@gxcme.edu.cn
[2] Research Fellow, Centre for Innovation and Development, Nanjing University of Science and Technology, Nanjing, China, e-mail: taolee@njust.edu.cn
[3] Associate Professor, Senior Eng., School of Artificial Intelligence Technology, Guangxi Technological College of Machinery and Electricity, Nanning, China, e-mail: 22605028@qq.com
[4]* Ph.D., Lecturer, School of Artificial Intelligence Technology, Guangxi Technological College of Machinery and Electricity, Nanning, China; *corresponding author, e-mail: duyingjun1984@126.com

to cloudburst attacks. In addition to the SHSE and Alipay incidents, cloudbursts have precedents in various stock exchanges across the globe, e.g., 2013 NASDAQ (New York), 2020 NZX (New Zealand), 2014 NSE (India), etc.

A cloudburst in RTCS can lead to immeasurable financial losses for traders and a technological credibility crisis [5]. We reconstructed the timeline of the SHSE incident (Fig. 1.): market dealers and traders experienced delays in matchmaking at the market open, followed by downtime of the exchange servers both in the AM and PM. This case not only exposed the architectural limitations of trading systems in high-pressure environments but also heightened market panic about the technology stableness and triggered public concern about the RTCS reliability.

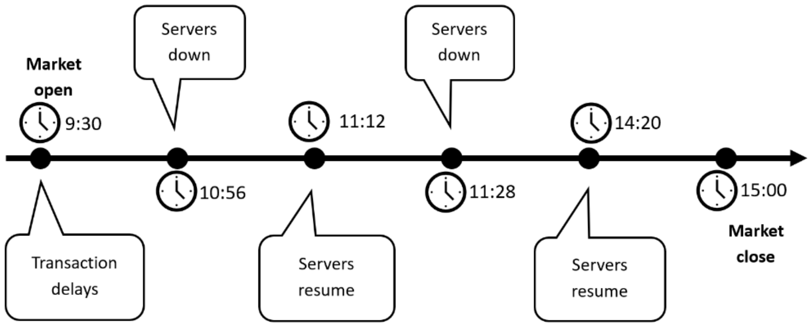


Fig. 1. Timeline of SHSE Cloudburst in Sep 27th, 2024

Architectural limitations are an inherent risk that makes it difficult for RTCS to withstand cloudbursts. The trading system and quotation system, as the core components of RTCS, rely on high-performance computing and data processing capabilities. The evolution of modern securities trading system architectures has resulted in subsystems such as trade matching, order communication, quote dissemination, and disaster recovery and backup working together in a distributed topology, with market dealers and traders accessing through data center LANs or WANs (Fig. 2.). The trading host cluster consists of two major platforms, BMP (Bid Matching Platform) and TP (Transaction platform). Wherein, TP includes ATP (Alternative Transaction Platform: trading services for diverse financial products), DTP (Derivatives Transaction Platform: trading options and derivatives), ITP (International Transaction Platform: cross-border securities trading services), XBTP (Extreme Bond Transaction Platform: trading and clearing services for newly issued bonds), FISP (Fixed Income Securities Platform), IITP (Internet Innovation Transaction Platform: online trading services). Taking SHSE as an example, it adopts streaming interface TDGW (trading gateway) with distributed architecture, a 100,000tps throughput, average latency less than 25 ms, high availability RPO (Recovery Point Objective) equals RTO (Recovery Time Objective) less than 30 seconds, with gateway level scalability, database polling for order reporting, message middleware adopting HDFS, and OpenVMS+Linux based on x86 server.

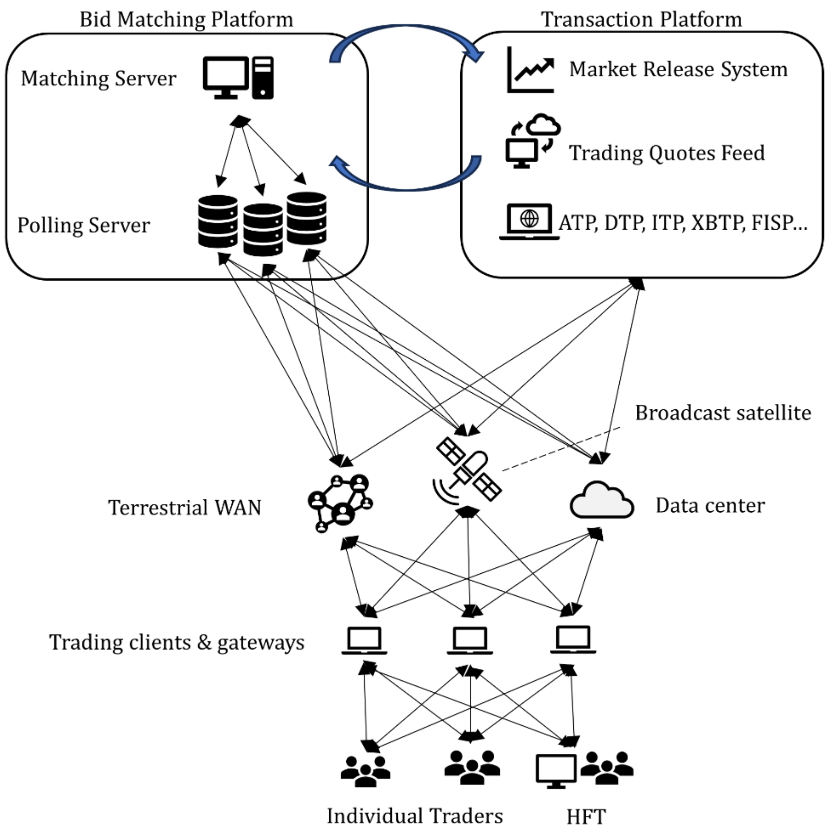


Fig. 2. RTCS Architecture

The BMP system comprises a matchmaking server and a polling server. The matchmaking server is responsible for processing and executing trade orders. When investors submit bids or asks, the server matches them based on price and time priority. Once a transaction is completed, a result is generated and sent to the polling server to update its data cache, which then transmits the order information and account balances back to the trader. The matchmaking server employs locking mechanisms on trading varieties and account funds to ensure data consistency. However, high-frequency trading may lead to lock contention[6,7]. Improperly designed locking mechanisms can increase the risk of deadlocks, posing an inherent risk of system downtime.

This study proposes a tech-neutral mechanism addressing RTCS cloudbursts by integrating market and cloud considerations to prevent high-frequency trading risks, avoiding the ethical dilemmas like the "Trolley Dilemma" associated with purely technical solutions[8-13]. The paper is structured as follows:

research background and contributions; recent related work; an evolutionary game model examining ESS results and simulations; and a summary with limitations.

2. Related Works

Cloud computing is widely adopted in modern smart financial and trading systems due to its flexible scaling and high availability features[14,15]. The problem of cloudbursts, i.e., the surge in cloud computing demand but the server cannot respond in time, has received attention from scholars [16]. Studies have shown that open-source hybrid clouds such as Apache Mesos can effectively address this problem[17]. In addition, heuristic scheduling algorithms have been used to explore performance and cost optimization of clouds[18]. Our previous research integrates architectural variants and incorporates agility factors such as capacity availability and cost into a quantitative cloud burst analysis framework[19].

In real-time trading, HFT (high-frequency trading) is one of the factors of EPM (extreme price volatility). Studies have demonstrated that high-frequency trading (HFT) plays a role in providing liquidity during event-driven price movements (EPMs), irrespective of whether these movements result in eventual price reversals or permanent price changes [20]. Concurrently, it has been argued that HFT exhibits characteristics more akin to speculative behavior rather than traditional [21]. The technical difficulties of monitoring HFT create regulatory challenges [22]. As a result, there is a lack of a mechanism to govern real-time trading that goes beyond purely technological means for the full penetration of HFT.

The rise in request concurrency necessitates addressing deadlocks, characterized by mutual exclusion, holding, no preemption, and loops [23]. Distributed database systems maintain consistency via concurrency control techniques like two-phase locking, timestamp ordering, and optimistic concurrency control [8]. These systems often balance key goals, trading off between consistency, availability, and fault tolerance [10]. Complex real-time relational databases can enhance query performance with similarity-based partitioning strategies [11]. The utilization of neutrosophic logic was proposed to enhance deadlock detection through the evaluation of transaction features and associated deadlines [9]. Furthermore, communication deadlocks were distinguished as a separate issue from resource deadlocks, with their identification being achieved through the implementation of multi-threaded testing methodologies [24].

Existing methods, such as performance simulation, were critiqued for their insufficient validation of schedulers, prompting the introduction of a novel methodology for scheduler description that specifically addresses deadlock fairness[13]. The complexity of deadlock identification was proposed to be reduced through the application of macro-operations [25]. A parallel deadlock detection algorithm was developed for heterogeneous platforms to enhance detection efficiency[12]. Research was conducted to determine the optimal thread pool size, focusing on system performance optimization [26]. Despite the growing interest in AI, concerns regarding dataset bias and its potential impact on fairness were raised. Rescheduling in production systems was advocated as an effective approach to resolving deadlocks [27], while anti-deadlock strategies were specifically designed for fog computing environments to improve system reliability [28].

Research on extreme price volatility driven by panic trading has extensively explored financial market behavior, behavioral finance, and market microstructure. A behavioral finance theory was introduced to address market bubble collapses, with a particular emphasis on the role of psychological factors [29,30]. To stabilize volatility, trading meltdowns or limits halt trading, with effects lasting up to 30 days [31]. However, the effectiveness of trading meltdowns has been debated, as it was suggested that they might adversely influence trading behavior when the quality of available information is inadequate [32]. Additionally, it was found that trading meltdowns significantly alter market dynamics and investor welfare, with threshold levels creating a "magnet effect" that influences market behavior [33].

Multi-modal AI agents (MAA) for natural interactions in physical and virtual spaces, highlights the integration of multisensory input, external knowledge, and human feedback. This embodiment helps AI agents perceive and adapt to environments, reducing errors in large models. The incorporation of common human knowledge across five domains—function, physics, intent, causation, and utility (FPICU)—has been emphasized as a critical framework for improving AI capabilities [34]. These studies offer new approaches for cognitive AI (CAI) adoption, aiming to enhance the ability of CAI to manage complex market environments and suppress trading panic.

To sum up, previous studies show justice issues with scheduling algorithms and AI enablement when RTCS experiences cloudbursting, and the meltdown mechanism is suspected of over-interfering with the market. Thus, the established approaches all affect investor welfare, and essentially, they rely on a single technology to try to solve complex system problems. This study thus attempts to fill that research gap by proposing a technology-neutral solution mechanism. In the following section, this study constructs a dynamic game of costs, benefits, and information on the market side and the cloud service side and employs evolutionary game theory and simulation to quantitatively discuss the systematic design of the mechanism.

3. The Mechanism and Game Analysis

This section introduces a technology-neutral mechanism for addressing RTCS cloudburst issues and analyzes its stability using evolutionary game theory. Technological neutrality is shaped by inherent value systems and ethical values [35,36]. The proposed mechanism avoids favoring specific process scheduling algorithms or cloud solutions in mitigating DDoS risks from RTCS cloudbursts. It aims to maintain transaction fairness and effectiveness by screening and delaying high-frequency Ask/Bid queue-jumping requests, thus preventing market distortion, shutdowns, or limited innovation. The ethical value lies in preserving justice, fostering innovation, preventing distortions, and ensuring transparency to support sustainable technological development and market competition.

Cloudbursts that lead to interruptions in service can bring about reputational damage and financial loss to traders, trigger failure contingency measures, compensation and regulatory penalties. The tech-neutral mechanism for preventing RTCS cloudburst takes into account the costs, benefits, and information asymmetries of considering both the market side and the cloud side (Fig 3).

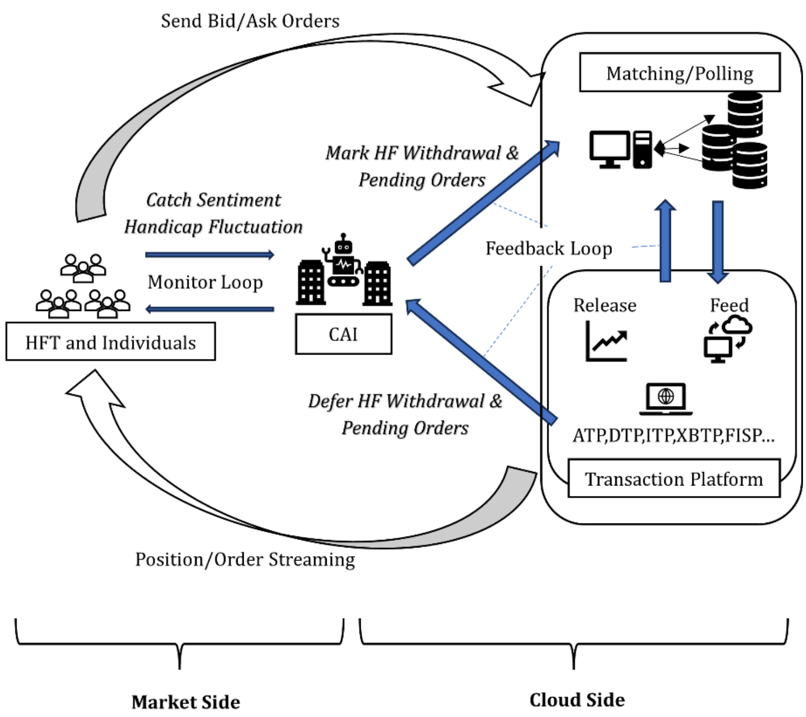


Fig. 3. Conceptual Framework of Tech-Neutral Mechanism

The mechanism is a dynamic gaming system that includes, in addition to market-side HFT traders and retail investors, cloud-side participants, i.e., real-time trading organizations, which provide infrastructure and services in terms of computational power, storage, and network resources, as well as brokerage firms, which provide Cognitive AI Agent (CAI) capabilities. Information sharing and signaling is a dynamic gaming process of the system, the market side submits pending orders and withdrawal signals, CAI monitors abnormal high-frequency pending orders and withdrawal signals and marks them and then transmits them to the cloud side, which assesses the risk of cloudbursts based on the received information and the current workload. If the risk of cloudbursts is high, the cloud side of the marked abnormal high-frequency pending orders and withdrawal signals for deferred polling and matchmaking. The cloud side thus avoids overload through dynamic strategies.

The Cognitive AI agent (CAI) is goal-oriented, facilitating dynamic human and management interactions, and is capable of autonomously acquiring knowledge across platforms for world model training and executing massive tasks (Fig. 4).

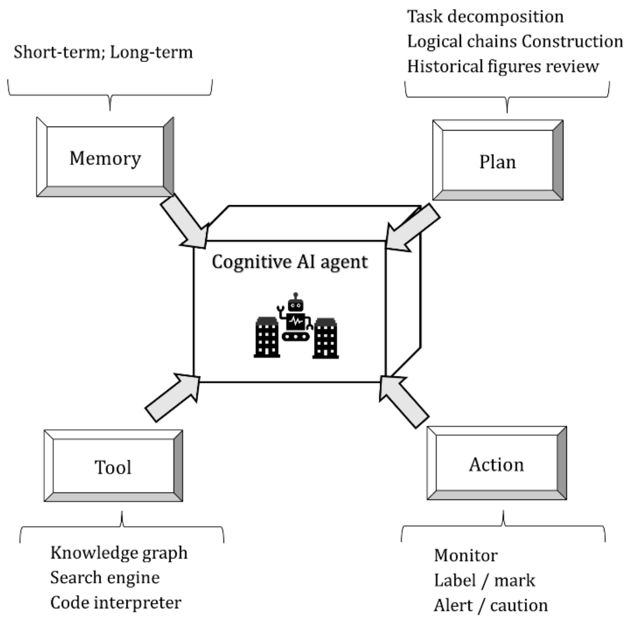


Fig. 4. Architecture of the Cognitive AI agent (CAI)

The architecture of CAI comprises four modules. In memory module, short-term memory is responsible for the big data memory of market volatility triggered by ad-hoc events, while long-term memory handles big data memory for (quasi-)periodic timeframes. Tool module includes knowledge graph, search engine, and code interpreter. Plan module encompasses task decomposition, the construction of logical chains, and the review of historical figures. Action module is tasked with monitoring, labeling, and alerting regarding high-frequency trading instructions in the market.

The technology-neutral mechanism we propose can avoid the Trolley Dilemma. The main ethical contradiction of the Trolley Dilemma lies in the empirical judgment based on utilitarianism, i.e., people choose to sacrifice the minority or the majority to solve the apparent problem, instead of preventing and solving the crisis by looking at the root cause of the problem. As a result, it raises controversies about rationality and fairness. For example, the difference in neural mechanisms between deontological and utilitarian judgments in moral dilemmas was revealed through the MERDJ neural computational model, but the oversimplified decision-making model (ignoring culture, experience, and situational dynamics) and the reliance on artificial simulation of dilemmas make it difficult to replicate real moral complexity [37]. Selfish or altruistic decisions are often influenced by the values that individuals in a dilemma place on the decision maker, and positive or negative emotions generated by facing moral dilemmas will distort moral judgments [38]. This suggests that relying on human subjective judgments or attempting to replicate moral judgments algorithmically is not appropriate in high-frequency problems. In the process of solving the RTCS cloudburst problem, we found that traditional technical approaches and algorithms, such as "kill locks", can release resources but lead to unexpected termination. Existing algorithms are unable to self-verify their fairness and thus raise similar ethical dilemmas: sacrificing the interests of a few traders to ensure the overall operation of the system.

The mechanism proposed in this study systematically circumvents the Trolley Dilemma through tech-neutrality and high-frequency trading monitoring. Tech-neutrality refrains from favoring specific process scheduling algorithms or cloud computing technical solutions, thereby preventing market distortions or unfairness arising from technological biases. This ensures the fairness and transparency of the mechanism and avoids ethical dilemmas induced by technological choices. High-frequency trading monitoring deploys CAI on both the market and cloud service sides to detect and flag abnormal high-frequency trading instructions. It dynamically assesses cloudburst risks and balances system load through deferred polling and matching, thereby safeguarding system stability while preserving transactional fairness.

3.1 Model Assumptions

Assumption 1: Traders are limited rational and act with the goal of maximizing short-term profits. The trader's strategy is {P - panic, H - hold} and the cloud strategy is {T - trigger, NT -not trigger}. The trader acts first and the cloud acts later. This dynamic game process is shown in Fig 5.

A variety of factors can trigger traders to {P}, which manifests itself in OB volume fluctuations. The fluctuation is not only from off-exchange trades into the on-exchange, but also from OB orders that choose to withdraw and then update the price to enter the OB waiting to be filled in order to accelerate the transaction. Such fluctuations aggravate the resource-intensive operations of locking, writing, and releasing the traders' account balances during the polling of the cloud servers.

Assumption 2: At time t, the proportion of traders choosing {P} is $x \in [0,1]$. The probability that the cloud takes {T} is $y \in [0,1]$.

Assumption 3: The proportion of high perception traders is denoted as θ and low perception as 1-θ. High perception traders prefer {P}. The public sentiment diffusion in the securities market prompts more traders to perceive the scarcity of trading opportunities, which in turn provokes a large number of traders to trade against the attempts to follow the OB price changes, which is the cause of cloudbursts.

Assumption 4: A high perception trader chooses {P} and realizes that the likelihood of misjudge is x(1-θ). The likelihood that a low-perception trader {H} later realizes that he or she missed a gain opportunity is (1-x)θ.

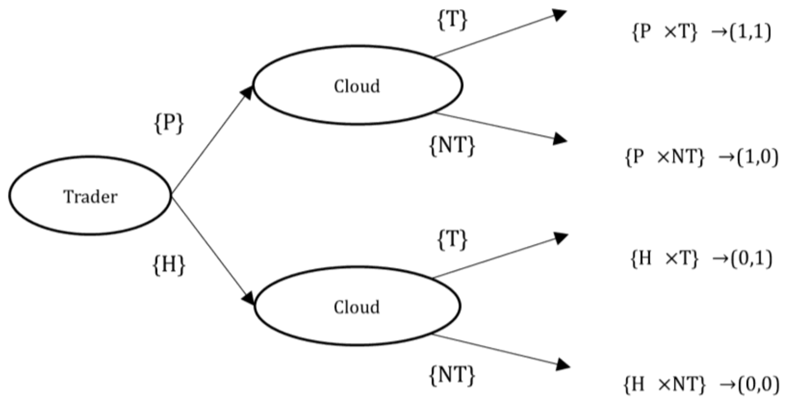


Fig. 5. The Dynamic Game Process

3.2 Payoff Matrix and RDE

We organize the symbols and descriptions related in the following section into Table 1.

Table 1

Related Symbols and Parameters

Symbol	Type	Description
B_1	proportional	Favorable impact of triggering the tech-neutral mechanism to maintain normal trading

C_1	economical	Cost of early warning cloudbursts
B_2	economical	Benefits of tech-neutral mechanisms triggered to avoid cloudbursts
C_3	economical	Regulatory penalties for the occurrence of cloudbursts
C_4	economical	Remediation costs in the occurrence of cloudbursts
C_5	economical	Reputational damage to the exchange as a result of the occurrence of cloudbursts
p	economical	Average OB price for panic trading
v	economical	Volume of orders accumulated at OB during panic trading
m_L	economical	Perceived value to OB by low perception traders
m_H	economical	Perceived value to OB by high perception traders
α	proportional	The extent of market facilitation when triggering tech-neutral mechanism
θ	proportional	Proportion of high perception traders
x	proportional	The proportion of traders choosing {P - panic}
y	probabilistic	The probability of a tech-neutral mechanism {T - trigger}

Based on preceding described game information, interactions between trader and cloud yield the following payoff matrix (Table 2).

Table 2

Payoff Matrix

			Cloud	
			{T} y	{NT} $1-y$
Trader	High Perception θ	{P} x	$\alpha B_1C_1+m_H-pv$ $B_2-\alpha C_1-C_4$	m_H-pv $-C_4-C_3-C_5$
		{H} $1-x$	B_1C_1 $-C_1$	0 0
	Low Perception $1-\theta$	{P} x	$\alpha B_1C_1+m_L-pm$ $b-\alpha c-s$	m_L-pm $-C_4-C_3-C_5$
		{H} $1-x$	B_1C_1 $-C_1$	0 0

The expected return under the trader's {P} action is denoted by π_{t1} = -pv + $y\alpha B_1C_1+\theta m_H+m_L$-$\theta m_L$. The expected return under the trader's {H} action is denoted by $\pi_{t2}=yB_1C_1$. Hence, the trader's average expected return between actions {P} and {H} is denoted as $\bar{\pi}_t=x\pi_{t1}+(1\text{-}x)\pi_{t2}$.

The expected return under the cloud {T} action is denoted as $\pi_{c1}=xB_2+(-1+x\text{-}x\alpha)C_1\text{-}xC_4$. The expected return under the cloud {NT} action is denoted as

π_{c2} = -x($C_3 + C_4 + C_5$). Hence, the average expected return of the cloud between {T} and {NT} actions is denoted as $\bar{\pi}_c = y\pi_{c1} + (1\text{-}y)\pi_{c2}$.

In the RTCS evolutionary game, the trader action probability and the cloud action probability are time-varying. Thus, derivatives to t yields RDEs: $\varphi(x) = \frac{dx}{dt} = x(\pi_{t1} - \bar{\pi}_t)$ and $\phi(y) = \frac{dy}{dt} = y(\pi_{c1} - \bar{\pi}_c)$. A further expansion as below.

$$\varphi(x) = (-1+x)x\left[pv - y(-1+\alpha)B_1C_1 - \theta m_H + (-1+\theta)m_L\right] \quad (1)$$

$$\phi(y) = (1-y)y\left[xB_2 + (-1+x-x\alpha)C_1 + x(C_3+C_5)\right] \quad (2)$$

Let (1)=0 and (2)=0, respectively, to obtain five evolutionary equilibriums: $(0,0),(0,1),(1,0),(1,1),(x^*,y^*)$, in which, $x^* = -\frac{C_1}{-B_2-C_1+\alpha C_1-C_3-C_5}$, and $y^* = \frac{pv\text{-}\theta m_H+(\text{-}1+\theta)m_L}{(\text{-}1+\alpha)B_1C_1}$.

The Jacobian Matrix obtained through partial derivatives as below.

$$J = \begin{vmatrix} \frac{\partial\varphi(x)}{\partial x} & \frac{\partial\varphi(x)}{\partial y} \\ \frac{\partial\phi(y)}{\partial x} & \frac{\partial\phi(y)}{\partial y} \end{vmatrix} \quad (3)$$

The five evolutionary equilibriums were brought into (3), to obtain the eigenvalues. Use the Lyapunov method[39] for determination based on the eigenvalues of Table 3.

Table 3

Eigenvalues of Equilibrium Points

Equilibriums	λ_1	λ_2
$E_1(0,0)$	$pv-\theta m_H-(1-\theta)m_L$	$B_2-\alpha C_1+C_3+C_5$
$E_2(0,1)$	$-\left[pv-\theta m_H-(1-\theta)m_L\right]-(1-\alpha)B_1C_1$	C_1
$E_3(1,0)$	$pv-\theta m_H-(1-\theta)m_L+(1-\alpha)B_1C_1$	$-B_2+\alpha C_1-C_3-C_5$
$E_4(1,1)$	$-pv+\theta m_H+(1-\theta)m_L$	$-C_1$
$E_5(x^*,y^*)$	*0*	*0*

3.3 ESS and Simulation

The methodology in this section is grounded in evolutionary game theory and further implemented through simulation and data science visualization. Eigenvalues provide the determination of ESS (evolutionary stable strategies) and its fulfillment conditions. A random number generator iteratively produces arrays that satisfy the conditions of RDE (replicated dynamic equations) and ESS, and these simulated arrays, represented as vector field flows, are displayed in phase plane vector field diagrams to analyze the formation of ESS. RDEs simulate the evolutionary behavior of the system, while the visualization of vector field flows is employed to identify points of convergence and divergence.

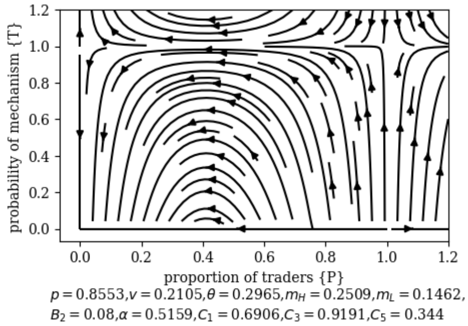
By observing that the eigenvalues obtain a determination of the ESS (Table 4), the emergence of ESS is divided into the following three theoretical scenarios. We employed vector field flow simulations to model the phase transitions of the ESS (Fig. 6). In the vector field flow, the horizontal and vertical axes represent the decision probability of the trader's {P} action and the activation probability of the tech-neutral mechanism {T}, respectively, both ranging within the interval [0,1]. The continuous game-theoretic interactions between the two participants are manifested as a dynamic convergence process within the vector field flow. The coordinate axes are displayed within the range of 0 to 1.2 to facilitate the observation of the vector field flow at the boundary points (0,1), (1,1), and (1,0). The flow of the vector field is depicted using arrowed curves. For instance, in Fig. 6a, the vector field flow converges at (0,0), diverges at (0,1), diverges at (1,0), and diverges at (1,1), thereby intuitively indicating that the ESS corresponds to $E_1(0,0)$. The ESS observation method remains consistent across the remaining subfigures. The simulations in this section were conducted based on the following hardware and software configuration: CPU, 13th Gen Intel(R) Core(TM) i7-1360P 2.20 GHz; RAM, 16GB; Operating system, Windows 11 24H2; Architecture, 64bit; Python version, 3.11.10.

In condition 1 and condition 3, $E_1(0,0)$ is the unique evolutionarily stable equilibrium (Fig 6a, Fig6d), in which {H, NT} is the normal state of market trading and no need for a triggered mechanism. Condition 2 yields two evolutionarily stable equilibrium points, $\mathrm{E}_3(1,0)$ and $\mathrm{E}_4(1,1)$, i.e., strategies combinations of {P × NT} and {P × T} (Fig 6b, Fig 6c).

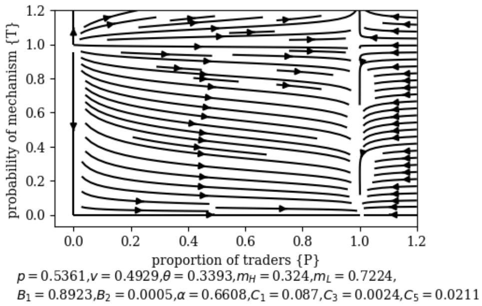
Table 4

Determination of ESS

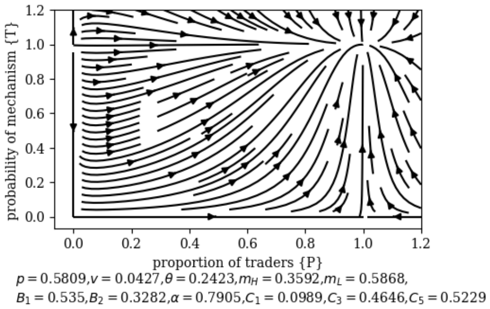
Equilibriums	*Condition 1*			*Condition 2*			*Condition 3*		
	λ_1	λ_2	**result**	λ_1	λ_2	**result**	λ_1	λ_2	**result**
$E_1(0,0)$	*neg*	*neg*	ESS	*pos*	*pos*	S	*neg*	*neg*	ESS
$E_2(0,1)$	*pos*	*pos*	S	*pos*	*pos*	S	*neg*	*pos*	N
$E_3(1,0)$	*pos*	*neg*	N	*neg*	*neg*	ESS	*pos*	*pos*	S
$E_4(1,1)$	*pos*	*pos*	S	*neg*	*neg*	ESS	*pos*	*neg*	S
$E_5(x^*,y^*)$	*0*	*0*	N	*0*	*0*	N	*0*	*0*	N



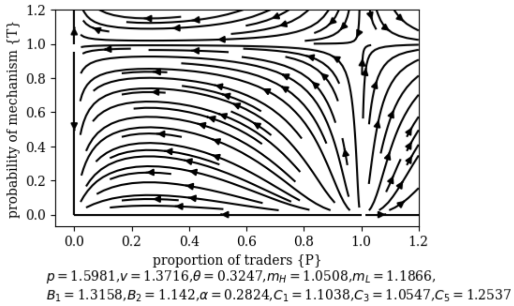
(6a) ESS $\mathrm{E}_1(0,0)$ under condition 1



(6b) ESS $\mathrm{E}_3(1,0)$ under condition 2



(6c) ESS $\mathrm{E}_4(1,1)$ under condition 2



(6d) ESS $\mathrm{E}_1(0,0)$ under condition 3

Fig 6 Phase Plane Vector Field Flow

Along the "benefit→cost→market facilitation" sequence, we conduct sensitivity analysis of the ESSs shift, from {P × NT} to {P × T}, in condition 2 using stepwise control variable method.

(1) Start point: {P × NT}. Simulation (Fig 7) shows the ESS appears to be $\mathrm{E}_3(1,0)$ when the benefit is low ($\mathrm{B}_1, \mathrm{B}_2 = 0.1$). While the benefit is high, the system is in an unstable state, thus control variables for further analysis are imposed in below steps (2) ~ (5).

(2) Given a low benefit level ($\mathrm{B}_1, \mathrm{B}_2 = 0.1$), observe the ESS convergence at various cost level $C_1, C_3, C_5 = 0.1,0.3,0.5,0.7,0.9$. Simulations show the mechanism hardly triggered (Fig 8).

(3) Tune to high benefit level ($\mathrm{B}_1, \mathrm{B}_2 = 0.9$) and observe the convergence of ESS at various level of the cost parameter (Fig 9). Simulations show traders tend to {P} action while the mechanism {T} action is unstable.

(4) Based on above simulation results, given high benefits and low costs, with various extent of market facilitation $\alpha = 0.1,0.3,0.5,0.7,0.9$, the simulation shows the mechanism tends to {T} faster under higher market facilitation (Fig 10).

(5) The stability of the ESS {P × T} obtained in the fourth step is verified by tuning the control variables to high benefit and high cost, when the mechanism is triggered with various extent of market facilitation (Fig 11). The simulation shows when the cost of the mechanism being triggered is elevated, the decrease in

net returns implies an increase in the difficulty of the mechanism chooses action {T}, which is public knowledge, and thus higher market facilitation stimulates traders to {P} action instead, at which point we find that the right subfigure of Fig. 11 shows the mechanism {T} state is unstable.

To summarize, the five-step control variable simulation described above shows that the high benefit and low cost of the mechanism facilitates the generation of {T} strategies, and that higher market facilitation allows the mechanism to be triggered more quickly, contributing to the stabilization of the system at a combination of {P × T}.

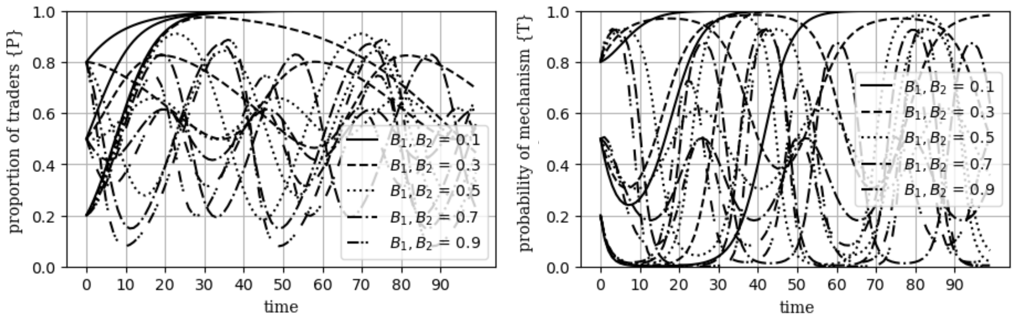


Fig 7 Impact of changes in benefit parameters on ESS

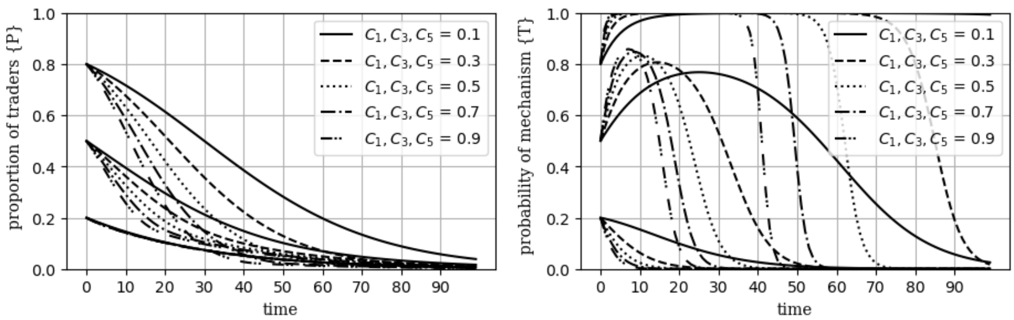


Fig 8 Impact of changes in cost parameters on ESS given low benefit

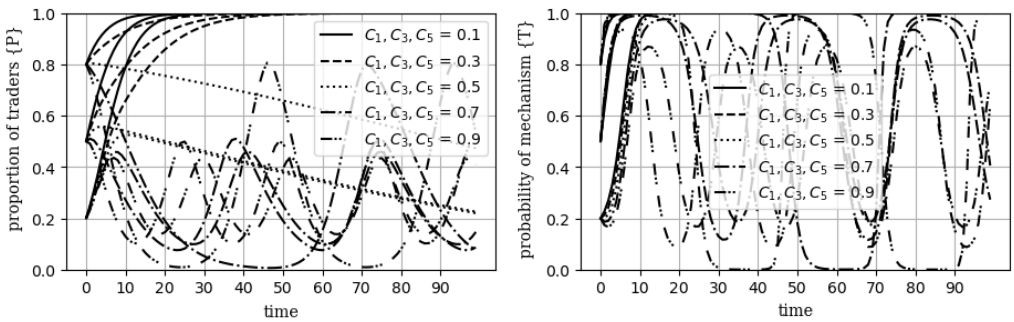


Fig 9 Impact of changes in cost parameters on ESS given high benefit

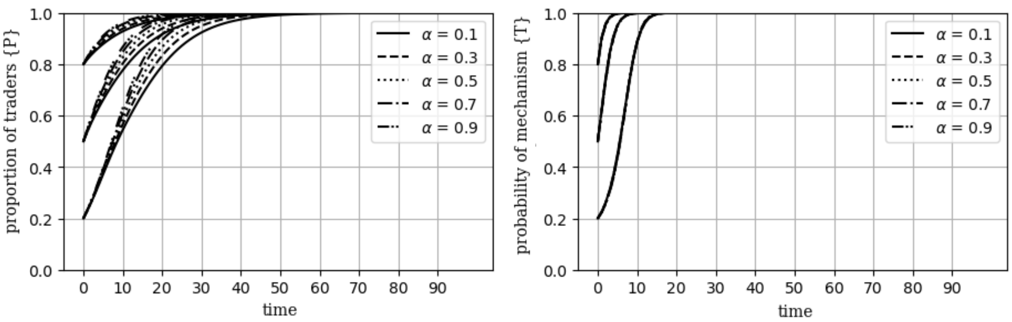


Fig 10 Impact of market facilitation on ESS given high benefit and low cost

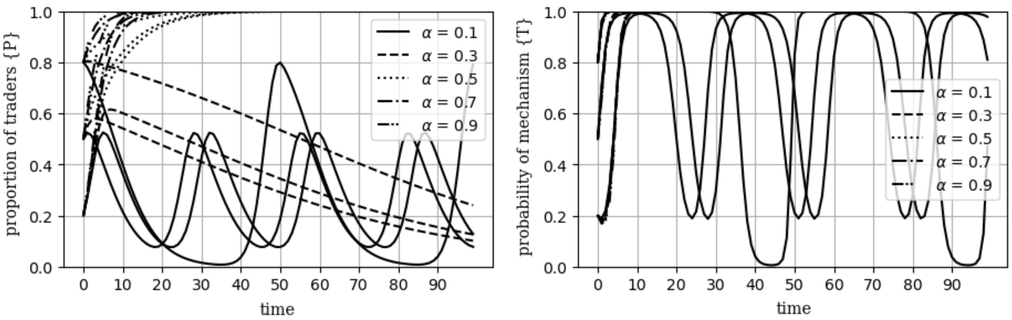


Fig 11 Impact of market facilitation on ESS given high benefit and high cost

4. Conclusion

With the development of 5G+ infrastructure and the enhancement of programmatic trading technology, it will become increasingly common for Real-time Trading Cloud-computing System (RTCS) to encounter cloudbursts. There are precedents from various stock exchanges across the globe, and the recent SHSE downtime and Alipay downtime during the “Double Eleven” e-commerce shopping festival continue to warn of the risk of cloudbursts. We point out the complex systemic causes of cloudbursts occurring at RTCS:

High-frequency pending and withdrawing requests send commands to the polling server through the trading terminal, and the matchmaking server matches orders based on price and time priority. While the cloud service is processing these high-frequency requests, the high-frequency requests in the queue are still undergoing withdrawals and pending order changes, which in turn leads to resource lock competition triggering deadlocks, which in turn triggers cloudbursts.

Through reviewing related studies, we find that the established process scheduling algorithms lead to the “Trolley Dilemma”, resulting in false kills and injustice, and the elastic scaling technique is only a passive technical strategy to cope with the problem. The fairness problem is still not solved.

Based on this, this study proposes a tech-neutral mechanism that suggests the introduction of cognitive AI agent (CAI) proactive defense to screen high-frequency requests located in the OB and deferred by the cloud-side polling and matchmaking servers, thus preventing cloudbursts while taking fairness into account. CAI has an important role in the tech-neutral mechanism of suppressing trading panic. Through multimodal perception and analysis, real-time monitoring and early warning, intelligent decision support, emotion management and psychological intervention, multimodal interaction and user experience, continuous learning and self-optimization and other multifaceted technological means, CAI can better cope with market fluctuations and reduce the occurrence of trading panic.

Acknowledgement

[1] “Research on User Profiling and Behavioral Modeling of Industrial College Education Based on Big Data Platform Featuring Industrial Meta-Universe Industry-Academia-Research with Panoramic Virtual-Realistic Integration”, #2022IT042, Zemin Zhang, Guangxi Technological College of Machinery and Electricity.

[2] 2024 Key Project of University-level Teaching Reform Research: Research and Practice of Teaching Mode of College General Course on Artificial Intelligence Based on Outcome-Based Education, 2024JY006, Yingjun Du, Guangxi Technological College of Machinery and Electricity.

REFERENCES

[1] *K. Alshouiliy& D. P. Agrawal*, Confluence of 4g LTE, 5g, fog, and cloud computing and understanding security issues, Springer Nature Switzerland AG, Cham,Switzerland, 2021.

[2] *H. Moudoud, L. Khoukhi& S. Cherkaoui*, "Prediction and detection of FDIA and DDoS attacks in 5g enabled IoT", in IEEE Network**, vol. 35**, no. 2, 2021, pp.194-201

[3] *F. Alhaidari, A. Rahman& R. Zagrouba*, "Cloud of things: architecture, applications and challenges", in Journal of Ambient Intelligence and Humanized Computing**, vol. 14**, no. 5, 2023, pp.5957-5975

[4] *H. Øverby, J. A. Audestad& H. Øverby, et al,* "Lock-in and switching costs", in Introduction to Digital Economics: Foundations, Business Models and Case Studies, 2021, pp.177-192

[5] *A. A. G. H. Nabil*, "Government cloud computing and national security", in Review of Economics and Political Science**, vol. 9**, no. 2, 2020, pp.116-133

[6] *M. Brolley& M. Zoican*, "On-demand fast trading on decentralized exchanges", in Finance Research Letters**, vol. 51**, 2023, pp.103350-103357

[7] *M. Salaheldin Elsayed, P. El Kafrawy& N. Wu*, "Modeling and deadlock control of reconfigurable multi-unit resource systems", in IEEE Access**, vol. 8**, 2020, pp.133605-133621

[8] *T. Nakamori, J. Nemoto& T. Hoshino, et al*, "Scalable timestamp allocation for deadlock prevention in two-phase locking based protocols", in Journal of Information Processing**, vol. 31**, 2023, pp.365-374

[9] *M. H. Hassan, S. M. Darwish& S. M. Elkaffas*, "An efficient deadlock handling model based on neutrosophic logic: case study on real time healthcare database systems", in IEEE Access**, vol.**

10, 2022, pp.76607-76621
[10] *J. Dizdarevic, Z. Avdagic& F. Orucevic, et al*, "Advanced consistency management of highly-distributed transactional database in a hybrid cloud environment using novel r-TBC/RTA approach", in Journal of Cloud Computing**, vol. 10**, no. 1, 2021, pp.27-58
[11] *S. Tarun, R. S. Batth& S. Kaur*, "A novel fragmentation scheme for textual data using similarity-based threshold segmentation method in distributed network environment", in International Journal of Computer Networks and Applications**, vol. 7**, no. 6, 2020, pp.231-252
[12] *E. E. Ugwuanyi, M. Iqbal& T. Dagiuklas,* "A comparative analysis of deadlock avoidance and prevention algorithms for resource provisioning in intelligent autonomous transport systems over 6g infrastructure", in Ieee Transactions On Intelligent Transportation Systems, **vol. 24**, no. 7, 2022, pp.7444-7461
[13] *M. Soualhia, F. Khomh& S. Tahar*, "Failure analysis of hadoop schedulers using an integration of model checking and simulation", in Electronic Proceedings in Theoretical Computer Science**, vol. 342**, no. 2021, pp.114-128
[14] *S. Vinoth, H. L. Vemula& B. Haralayya, et al*, "Application of cloud computing in banking and e-commerce and related security threats", in Materials Today : Proceedings**, vol. 51**, no. 2022, pp.2172-2175
[15] *M. Cheng, Y. Qu& C. Jiang, et al*, "Is cloud computing the digital solution to the future of banking?", in Journal of Financial Stability**, vol. 63**, no. 2022, pp.101073-101093
[16] *L. Chen& W. Chang*, "Under what conditions can an application service firm with in-house computing benefit from cloudbursting?", in European Journal of Operational Research**, vol. 282**, no. 1, 2020, pp.71-80
[17] *H. Haugerud, N. Xue& A. Yazidi*, "On automated cloud bursting and hybrid cloud setups using apache mesos", in Concurrency and Computation: Practice and Experience**, vol. 32**, no. 17, 2020, pp.e5662
[18] *G. L. Stavrinides& H. D. Karatza*, "Cost-aware cloud bursting in a fog-cloud environment with real-time workflow applications", in Concurrency and Computation: Practice and Experience**, vol. 33**, no. 23, 2021, pp.e5850
[19] *J. Cheng& T. Li*, "Agility factors analysis for financial platform enterprises under cloudbursting arrivals", in UPB Scientific Bulletin, Series C: Electrical Engineering and Computer Science**, vol. 85**, no. 4, 2023, pp.47-58
[20] *J. Brogaard, A. Carrion& T. Moyaert, et al*, "High frequency trading and extreme price movements", in Journal of Financial Economics**, vol. 128**, no. 2, 2018, pp.253-265
[21] *A. Roncella& I. Ferrero,* "The ethics of financial market making and its implications for high-frequency trading", in Journal of Business Ethics, **vol. 181**, no. 1, 2022, pp.139-151
[22] *V. Manahov,* "High‐frequency trading order cancellations and market quality: is stricter regulation the answer?", in International Journal of Finance & Economics, **vol. 26**, no. 4, 2021, pp.5385-5407
[23] *Z. Qiu, S. Shao& Q. Zhao, et al*, "A characteristic study of deadlocks in database-backed web applications," in 2021 IEEE 32nd International Symposium on Software Reliability Engineering (ISSRE), 2021, pp.510-521
[24] *D. Khanna, R. Purandare& S. Sharma*, "Synthesizing multi-threaded tests from sequential traces to detect communication deadlocks," in 2021 14th IEEE Conference on Software Testing, Verification and Validation (ICST), 2021, pp.1-12
[25] *Y. Huang, T. Wang& Z. Yin, et al*, "Improving the efficiency of deadlock detection in MPI programs through trace compression", in IEEE Transactions On Parallel and Distributed Systems**, vol. 34**, no. 1, 2023, pp.400-415
[26] *D. Casini*, "A theoretical approach to determine the optimal size of a thread pool for real-time systems," in 2022 IEEE Real-Time Systems Symposium (RTSS), 2022, pp.66-78

[27] *A. L. De Sousa& A. S. De Oliveira*, "Finite-time consensus and readjustment three-stage filter for predictive schedules in FMS", in IEEE Access**, vol. 11**, no. 2023, pp.88558-88582

[28] *M. Talaat, A. Saleh& M. Moawad, et al*, "Fog computing effective load balancing and strategy for deadlock prediction management", in Ain Shams Engineering Journal**, vol. 14**, no. 12, 2023, pp.102561-102572

[29] *S. Khan, H. Goyal& V. K. Meena,* "Emerging trends in behavioral finance: a bibliometric analysis", in Iup Journal of Financial Risk Management, **vol. 21**, no. 2/3, 2024, pp.55-71

[30] *T. Hon, M. Moslehpour& K. Woo,* "Review on behavioral finance with empirical evidence", in Advances in Decision Sciences, **vol. 25**, no. 4, 2021, pp.1-30

[31] *C. A. Charles& M. S. Kulkarni*, "Circuit breaker effects on market players psychology and behavior", in Cardiometry**, vol. 1**, no. 23, 2022, pp.244-254

[32] *J. Magnani& D. Munro*, "Dynamic runs and circuit breakers: an experiment", in Experimental Economics**, vol. 23**, no. 1, 2020, pp.127-153

[33] *H. Chen, A. Petukhov& J. Wang, et al*, "The dark side of circuit breakers", in The Journal of Finance**, vol. 79**, no. 2, 2024, pp.1405-1455

[34] *Y. Zhu, T. Gao& L. Fan, et al*, "Dark, beyond deep: a paradigm shift to cognitive AI with humanlike common sense", in Engineering**, vol. 6**, no. 3, 2020, pp.310-345

[35] *B. Miller,* "Is technology value-neutral?", in Science, Technology, & Human Values**, vol. 46**, no. 1, 2021, pp.53-80

[36] *S. Heyndels*, "Technology and neutrality", in Philosophy & Technology**, vol. 36**, no. 4, 2023, pp.75-97

[37] *T. Gothard& J. Davies*, "A spiking neuron model of moral judgment in trolley dilemmas", in Scientific Reports**, vol. 14**, no. 1, 2024, pp.21733-21745

[38] *F. Liang, Q. Tan& Y. Zhan, et al*, "Selfish or altruistic? The influence of thinking styles and stereotypes on moral decision-making", in Personality and Individual Differences**, vol. 171**, 2021, pp.110465-110471

[39] *A. M. Lyapunov*, "The general problem of the stability of motion", in International Journal of Control**, vol. 55**, no. 3, 1992, pp.531-534