EAI Endorsed Transactions
on Scalable Information Systems

Research Article EAI.EU

Cloud Data Integrity Verification Algorithm for Smart Accounting Informatization in Cloud Computing

Jun Zhang[1*], Liang Xian[2]

[1]Department of Business, Sichuan University Jinjiang College, Peng Shan, Sichuan 620860, China.
[2]School of Economics and Management, Chengdu Sport University, Chengdu, Sichuan 610041, China.

Abstract

INTRODUCTION: With the rapid development of cloud computing and intelligent technology, traditional accounting informatization systems are gradually migrating to cloud computing platforms, promoting financial management's automation and intelligence.
OBJECTIVES: However, data security in the cloud computing environment, especially the issue of data integrity, has become one of the core challenges for intelligent accounting informatization systems. Accounting informatization systems rely on the storage and processing of a large amount of financial, tax, and transaction data, and the integrity of these data is directly related to the financial transparency, compliance, and decision-making accuracy of an enterprise.
METHODS: Therefore, how to design an efficient and scalable cloud data integrity verification algorithm becomes the key to ensuring the reliability and security of intelligent accounting systems. This paper proposes a cloud data integrity verification algorithm for smart accounting informationization based on a cloud computing environment. The algorithm combines the efficient data processing capability of cloud computing with the automation requirements of smart accounting systems to ensure data integrity during storage, transmission, and processing through various data validation techniques such as digital signatures, hash algorithms, and blockchain-based distributed ledgers.
RESULTS: The algorithm design considers the dynamics and scalability of the cloud computing environment. It can realize fast, real-time data validation and integrity detection in a large-scale data environment. Through experimental verification, the algorithm proposed in this paper performs well in processing large-scale accounting data with high verification efficiency and accuracy.
CONCLUSION: Compared with traditional validation methods, the algorithm improves the speed of data validation. It enhances the system's ability to prevent data tampering and loss, providing a new cloud data integrity assurance scheme for intelligent accounting informatization systems.

Keywords: cloud computing; smart accounting; information technology; data integrity; verification algorithms; digital signatures; blockchain.

Received on 03 March 2025, accepted on 19 November 2025, published on 02 December 2025

doi: 10.4108/eetsis.8828

*Corresponding author Email: zhangjun20241121@163.com; jun_zhang001@outlook.com

1. Introduction

Technology support for the financial services industry. With the rapid development of information technology, especially the application of cloud computing, big data and artificial intelligence, the traditional accounting informatization system faces unprecedented challenges and opportunities [1,2]. As a crucial field in enterprise and social management, the development of accounting informatization not only promotes the improvement of economic efficiency but also facilitates globalization and digital transformation. In this process, cloud computing, as a new computing model, is widely used in accounting informatization due to its efficiency, scalability and cost-effectiveness [3]. However, the openness and distribution of cloud computing also bring problems such as data security and data integrity, especially in intelligent accounting systems, how to ensure the integrity and authenticity of cloud data has become a key issue to be solved [4].

Cloud computing smart accounting informatics systems usually rely on cloud platforms for data storage and processing, which cover a large amount of financial, tax, transaction and other critical information [5]. Therefore, data integrity is critical because any data loss or tampering may have far-reaching negative impacts on an organization's financial position, audit results, and compliance [6]. Especially in smart accounting informatization systems, automated data processing and decision-making mechanisms depend more on data quality [7]. Traditional accounting data integrity verification methods, such as techniques based on signatures, hash functions, or data backups, are effective but often limited to specific application scenarios or difficult to adapt to cloud environments distributed and dynamic characteristics [8, 9]. Therefore, it is particularly important to propose a new, more efficient and reliable verification algorithm for accounting data integrity in cloud computing.

Currently, the research on cloud data integrity verification mainly focuses on the security guarantee of data storage, data transmission and data access; however, with the progress of intelligent technology, how to realize real-time data verification and automated detection in intelligent accounting information systems has become a new research direction [10]. Especially for the financial service field, the intelligent accounting system under the cloud computing environment needs to respond quickly and automate accounting data processing and report generation, relying on real-time data exchange and sharing on cloud platforms, which makes the integrity checking of cloud data more complicated and challenging. The current generation of cloud-based smart accounting software is confronted by a host of security issues, with the main one being the need to ensure that the financial data, which is stored on and processed by several different cloud platforms, retains its integrity. Already though, certain powerful tools such as digital signatures, hash verification, and block chain-based ledgers have started to be applied to secure data, however, their use in real-time, automated accounting systems still very much restricted. This research suggests an algorithm that not only addresses these challenges but also combines effective cloud data processing with automated integrity verification, thereby ensuring that data is consistent throughout the various stages of storage, transmission, and processing. Additionally, the algorithm enhances system's fault tolerance and adaptability, thereby facilitating secure and reliable accounting operations in constantly changing cloud environments. In this context, how to design an efficient, scalable and adaptable cloud data integrity verification algorithm for smart accounting informationization in a cloud computing environment becomes the key to ensuring system reliability and compliance [11, 12].

This paper proposes a cloud data integrity verification algorithm for smart accounting informationization based on a cloud computing environment. In cloud-based accounting systems, financial data is frequently transmitted, updated, and stored across distributed servers. This exposes the data to integrity risks, including unauthorized modification, accidental loss, and malicious tampering. Traditional verification methods such as simple hash checks or data backups cannot ensure consistency across multi-tenant, dynamic cloud environments, and lack traceability for cross-node verification. Any compromise of data integrity may lead to inaccurate accounting records, incorrect financial reports, and unreliable audit results, highlighting the need for robust cloud-oriented verification mechanisms [13]. The algorithm combines the efficient data processing capability of cloud computing and the automation needs of smart accounting systems and aims to ensure the integrity of data during transmission, storage, and processing by utilizing a variety of data validation techniques such as digital signatures, hash algorithms, and blockchain-based distributed ledgers [14]. Meanwhile, the algorithm also considers the dynamics and scalability in the cloud computing environment. It can realize fast verification and real-time detection of large-scale data. In addition, this paper explores the feasibility and performance of the algorithm in practical financial service applications. It experimentally verifies the efficiency and accuracy of the algorithm in processing large-scale financial data.

With the continuous development of accounting informatization and cloud computing, the application of intelligent accounting systems will become increasingly widespread, and the guarantee of data integrity will also face higher requirements. Constructing an effective data integrity verification mechanism can guarantee the authenticity, accuracy and consistency of the data and enhance the transparency and trust of the system, thus providing more secure and reliable information

1.1 Research Contribution

- A proposal has been made for a cloud data integrity verification algorithm that combines digital signatures, hash algorithms, and blockchain-based distributed ledgers to confirm the reliability of accounting data in cloud settings.

- The algorithm caters to dynamic and large-scale cloud systems, allowing instant detection and confirmation of financial data.

- The implementation of a lightweight and scalable verification framework not only increases verification efficiency but also lowers the consumption of system resources.
- Tests done with the method indicate that it is highly accurate and adaptable, thus affirming its appropriateness for smart accounting information systems.

2. Related work

At the beginning of the 21st century, human-computer interaction development theory and demand management theory were gradually introduced into the research field of software development in China [15]. The book "Scrum Human-Computer Interaction Project Management" by American Schwaber, translated by Li Guobiao, is the first time to introduce the theory of human-computer interaction development methods in China. With the rapid development of China's domestic software industry, human-computer interaction development methods have also been widely recognized and applied [16]. At the same time, he believes that two stages of requirements determination and requirements review should be focused on. Domestic scholars' research on demand management tools is also important in the current theoretical circle [17]. This paper sorts out and summarizes some common tools and methods of current demand management work and compares and analyzes the characteristics, applicable conditions, advantages and disadvantages of RDT, Rational Analyst studio, DOORS and other work. The monograph written by the Bank of China Software Center in 2018 systematically refined and summarized the theory, content and methods of the bank's software engineering system [18]. Focuses on applying human-computer interaction modeling in the development of financial business software, summarizes the shortcomings and problems of traditional waterfall model development, and puts forward the core principles and methods of demand management under human-computer interaction development. Combined with the case of software project improvement of ICBC Software Development Center, we focus on the itemized demand management model [19]. In one significant research, a framework for accounting in the cloud was created which highlighted data security, processing efficiency, and system scalability. The current work enhances data reliability with the implementation of a cloud data integrity verification algorithm that allows secure, tamper-proof, and reliable smart accounting operations [20].

To sum up, financial services have penetrated many fields under the theory of human-computer interaction. The introduced method uses blockchain distributed ledgers, which are already in place, to provide a stronger data integrity guarantee in the accounting systems based on cloud. Through the preservation of an unalterable and authenticated record of all the financial transactions, the system makes it impossible for any unauthorized changes or alterations to go unnoticed. By doing this, the traditional idea of quantum-proof digital authentication in distributed systems is moreover supported and hence offering a secure and trustworthy way for the maintenance of the accounting data integrity over the cloud services. Cloud-based accounting systems not only need to provide data integrity but also have to be very fault-tolerant to be able to work continuously and reliably. The new method contributes to system strength by managing data verification duties in an effective way and allowing itself to vary according to the changes in cloud resources. This approach lessens the likelihood of system crashes at the time of higher loads or unforeseen interruptions and thus, allows the processing of financial data to be uninterrupted and the reliability of intelligent accounting systems to be protected. When the cloud is utilized for processing large-scale accounting data, one of the main problems that arise is the need for optimised distribution of tasks between storage and computing layers. The proposed solution of the algorithm allows for the cloud-edge style processing approach offered where the data verification tasks are distributed and processed concurrently in different cloud nodes. This allows for quick, real-time integrity checks, lowering latency and load on the system, thus, processing large-scale, and high-frequency accounting operations smoothly and without causing bottlenecks in processing. In addition to data integrity and fault tolerance, the cloud accounting systems also need systematic risk assessment through which security breaches would be prevented. The algorithm that is proposed relies on risk evaluation since it is constantly monitoring data consistency and logging anomalies in the cloud nodes. The identification of suspicious patterns or irregularities in the accounting transactions is done thus providing a structured mechanism for the early detection and the subsequent mitigation of potential vulnerabilities in the financial data processing. Behavior monitoring becomes an additional crucial benefit of integrity verification and risk assessment to cloud accounting systems. The algorithm being proposed follows and interprets the verification patterns, so it can detect and expose the abnormality or attempting to access the system which is suspicious. This behavior perception gives room for the early detection of bugs or settings that are not proper, thus the overall security is improved and the financial data operations within the cloud environment continue to be secure. Any financial services company wants to attract customers' attention, maintain customer loyalty, and gain lucrative profits. To achieve this goal, analysing the process of customer use of financial services is necessary.

3. Research Objective

This topic sees the development of financial service software as an opportunity, takes human-computer interaction as the main development idea, develops an AD module driver with the help of the Linux kernel, and mainly completes module initialization, opening, data reading and writing, interrupt, release, and unloading functions. Efficient communication with hardware is realized, which facilitates further software development [21].

3.1 Key Technologies of Financial Software Module-Driven Development

In the Linux operating system, device drivers play a very important role. They can not only control the process but also need to provide other applications with a program interface to interact with the device. From the application point of view, the hardware device is just a file because the device driver shields the details of the hardware, and the user uses the device through programming operations such as file processing. The drive structure is shown in Figure 1.

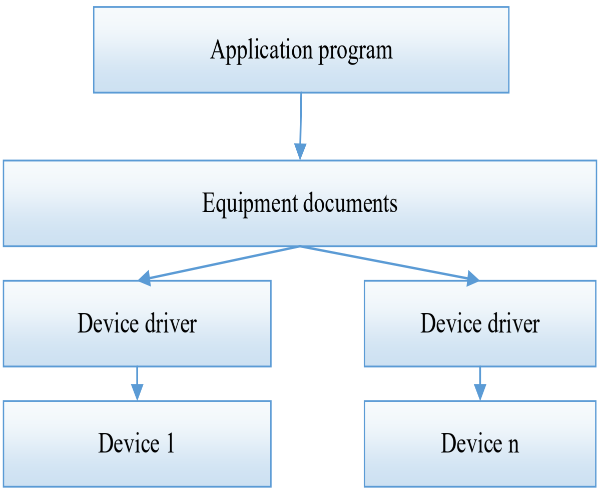

Figure 1. Schematic diagram of AD drive structure

There are two types of device files under the Linux financial service software operating system: character device, block device, and block device. As the name implies, the operating system will set a memory buffer area for the device. When users want to view the device, they only need to Detect the buffer first; if there is suitable data, directly fetch the data from this block of memory; if not, then call the I/O instruction to fetch the device data to the buffer. Character devices, not as complex as block devices, directly communicate with the device character by character for I/O, and the device will respond immediately every time data is read or written. Generally speaking, PCI devices are all character devices.

3.2 PCI bus configuration

Each device on the PCI bus has a data structure to configure, which belongs to the PCI configuration space. First, the PCI configuration header is the key for the operating system to identify the device. The location of this header is uncertain, and the operating system needs to perform a topology search for the device. For example, every slot on a PCI chassisBits is assigned a PCI configuration header location. When the card's location changes, the configuration header will appear elsewhere in the configuration space. Linux can query the location of the header and configure it through the status register. Generally speaking, the PCI structure makes the configuration headers of different slots have offset addresses related to their positions. The Linux kernel has a related mechanism for PCI devices, which can detect the configuration headers of all devices on the PCI bus. All correctly inserted and functioning PCI devices are available. Its PCI configuration space table is shown in Table 1.

Table 1. PCI configuration space table

Device ID		Vendor ID		00h
Status		Command		04h
Class Code			Revision ID	08h
BEST	Header Type	Latency Timer	Cache line Size	0Ch
Base address Re-registers				10h-24h
Cardbus Cis Pointer				28h
Subsystem ID		Subsystem Vendor ID		2Ch
Expansion ROM Base Address				30h
Reserved			Capabilities Pointer	34h
Reserved				38h
Max Lat	Min Gnt	Interrupt Pin	Interrupt Line	3Ch

A PCI device has up to 6 I/O address areas. The manufacturer sets the corresponding I/O registers in the memory space, and the configuration registers are marked with special markers.

3.3 Basic framework of financial service software

In the Linux kernel of financial service software, the device category needs to be described by a structure, and the variables defined by the structure can represent such specific devices. The variables have detailed information about this specific device. A complete driver can make Equipment of the same type that can be used. Otherwise, the Equipment cannot be mass-produced. Regarding the problem of distinguishing between Equipment of the same type, it is necessary to borrow the secondary equipment number. An array can be used to represent the Equipment of the same type, and a single element represents the specific Equipment, as follows As shown, this structure is designed to be used as a device class. Its interface diagram is shown in Figure 2:

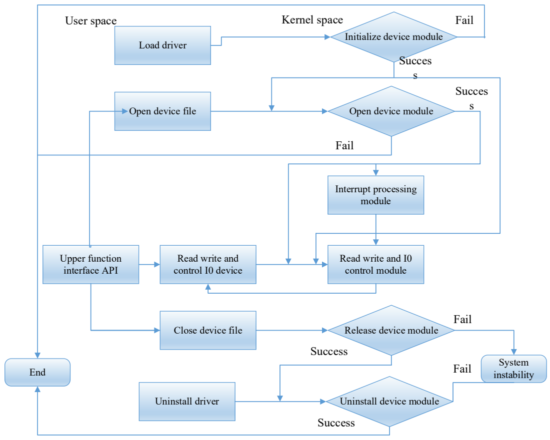

Figure 2. Structure interface diagram

Identifying important behavioral patterns in financial services software does not explicitly specify a method for identifying these behavioral patterns; Cooper believes that in previous output depictions, interviewees should group interview users by hypothesized human-computer interaction and then describe important user behavior variables middle. Its limitations are shown in Table 2 below.

Table 2. Cooper model human-computer interaction assignment limitations

Cooper model activities	Related limitations
Determine user behavior variables	It is not clear that the assumption has been completed, and the collected data has been roughly counted.
Describe interviewees to user behavior variables	The range of key user behavior variables was not determined
Identify important behavior patterns	It is not clear how to output these modes
Integrated features and related objectives	There is no output of the statute on comprehensive features and related objectives
Integrity and redundancy check	Instructions that do not indicate the process of performing this activity

4. Methods

It is very important for financial service software to reasonably choose an improved model of human-computer interaction technology. This study selects the human-computer interaction technology proposed by Cooper as the basis for improvement. Cooper was the originator of the original version of human-computer interaction technology and the basis for the research of other researchers. These different versions of human-computer interaction technology improved based on Cooper human-computer interaction technology have been successfully applied to a series of practical projects.

In the development process of financial service software, the T-CUM algorithm is an algorithm for calculating a single type of user behavior pattern. A single user behavior pattern is the basis for constituting the user behavior pattern, and the user behavior pattern is the skeleton that constitutes the role. After obtaining the user interview responses, the user types can be segmented according to the user's industry and occupation. After classification, the T-CUM algorithm can be applied to each subdivided user type to calculate the single-class user behavior pattern of the user. Such users' behaviours are first counted to obtain the total number of user behaviors. These are valid user behaviours if the proportion of user behavior exceeds 30%. While checking for the required user behavior, the number of user behavior variables is greater than 6. If both conditions are met simultaneously, calculate the percentage of the user behavior variable in the user behavior. Use the condition that the user behavior characteristics are greater than 10% to make judgments, get all the effective user behavior variables and their percentages, and list the user behavior variable graph to find the user behavior pattern. Figure 3 shows the algorithm flow chart of a single type of user behavior pattern.

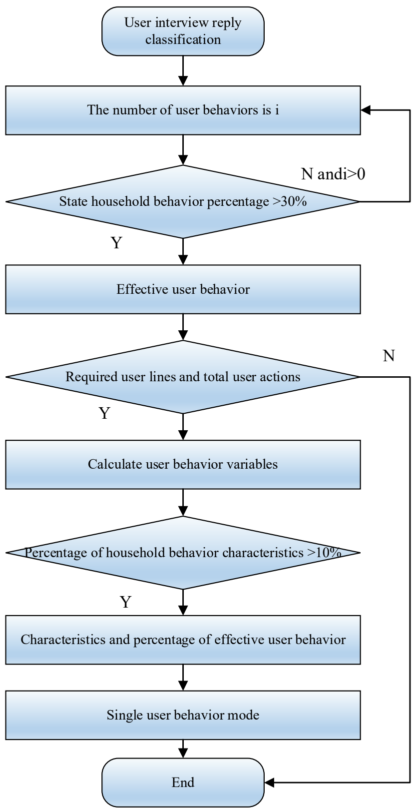

Figure 3. Flowchart of the algorithm of user behavior pattern

The hardware of the financial service software virtual assembly system developed in this subject can be abstracted into three functional modules, namely the display layer, the interaction layer and the data layer. Among them, the hardware of the display layer includes a ring projection screen, a multi-channel digital image fusion machine, an engineering projector, and three-dimensional glasses. The function of this layer module is to display a virtual image with three-dimensional immersion. Users can observe and experience the virtual experiment content by wearing the three-dimensional glasses. The hardware of the interaction layer includes the central control system, optical tracking device, and Leap Motion somatosensory controller. The function of this layer module is to collect the interaction information between the user and the virtual object in real-time, including the actions of the user's hand, such as clicking, grasping, and moving. The data layer hardware includes graphics workstations, video matrix switchers, and disk arrays. This layer is mainly responsible for processing the volume of data collected by the interaction layer and generating virtual images displayed on the screen in real-time, which is the basis for realising various system functions. The structural relationship of various hardware in the financial service software virtual assembly system is shown in Figure 4.

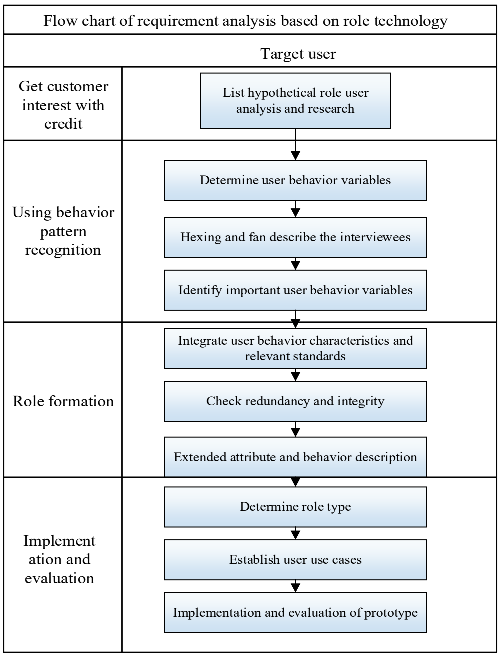

Figure 4. Structure diagram of virtual assembly system

5. Case study

In the process of financial service software development, in the program that uses the register_chrdev() function for driver registration, the first parameter represents the major number of its device driver, which is set to 0 here, and the Linux kernel assigns each device during the boot phase. Therefore, whenever the system state changes, the number will change accordingly, so the driver must solve this problem to ensure that the device is in the connected state without interruption [22, 23]. To solve the problem of changing the major device number of the device file corresponding to the AD module device during the loading process, it is necessary to create a correct device file for it first so that the system can access AD correctly [24]. The Linux kernel specifies /proc/devices. The cu. rrent device number is saved in the directory, the driver is loaded to query it, and finally, a corr,ect device file is newly generated by the read oureadoutor device number. As for DRVNAME as the driver name, it can be defined when the driver is compiled, generated, and set to AD here, as shown in Figure 5.

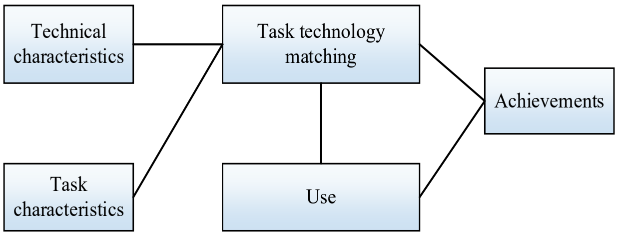

Figure 5. Empirical test of TPC

5.1 Improvement of Memory Allocation Efficiency of Financial Services Software Drivers

The most important task of the Linux device driver of the financial service software is to complete the device data transmission, which is realized by reading I/O and also needs to set the configuration space. It is necessary to allocate memory space in advance to facilitate resource access, and the AD driver needs to reside in the background of the system and some space [25, 26]. So, the driver itself also consumes a certain amount of memory. To ensure its memory consumption, the driver needs to allocate a certain amount of memory to itself when initialising the device. The weighted distribution diagram of its initialization efficiency is shown in Figure 6.

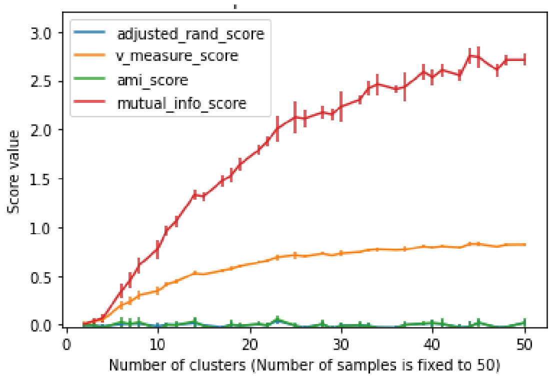

Figure 6. Initialization efficiency weighted allocation diagram

Like the calling method of malloc, use malloc to apply memory space for the AD driver and allocate it on demand. If it is too high, it will affect the system's running speed. For resources such as configuration space, I/O space, etc., to apply for memoryTo discuss in detail, the memory allocation mechanism under Linux is based on page length. If you want to apply for a continuous interval at a time, its length must satisfy the power of 2. The dimension diagram of its allocation mechanism is shown in Figure 7.

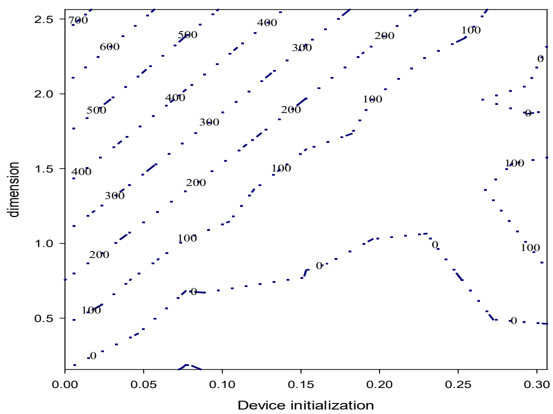

Figure 7. Dimensional diagram of the memory allocation mechanism

5.2 The opening speed of financial service software equipment modules is improved

The Linux kernel of the financial service software stipulates that the device driver opens the device, and the open() should be borrowed uniformly. It is in the DevAd item in the driver object defined in the AD module and needs to be implemented by the developer. If the driver wants to operate the device, it first opens it. The optimization flow chart is shown in Figure 8.

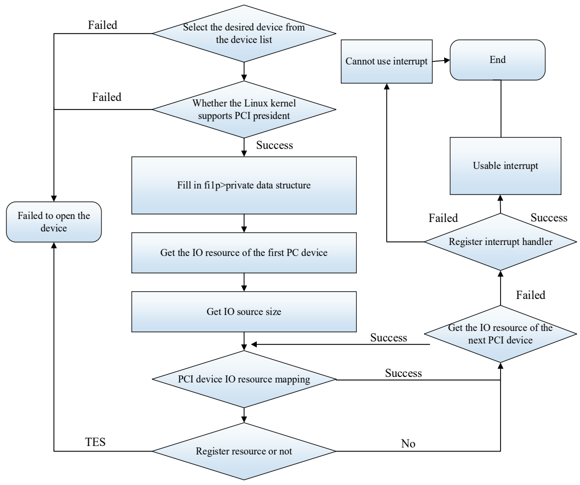

Figure 8. Flow chart of device initialization optimization

To take control of the device and open the device file, the driver must distinguish between two open methods: blocking and non-blocking. This AD module driver uses the semaphore down to determine the device's control, checks its value, and decrements its semaphore if it is greater than zero. If it is zero, the function is sleeping and needs to wait for other processes to wait for the next detection. Its loss function is shown in Figure 9.

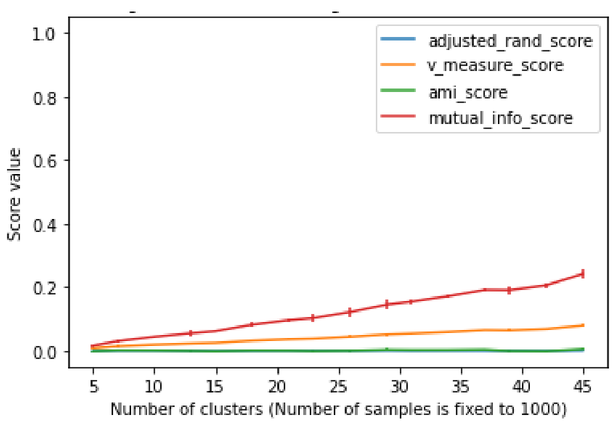

Figure 9. Convex loss function graph

In the financial service software AD module software, the I/O operation of the PCI device is extremely important. The I/O ports and I/O memory are collectively called the I/O area. In the PCI implemented by IPCORE used in this system, there are up to 6 I/O address divisions. This AD module only uses two. The I/O memory resource BAR0 is designed by the manufacturer to be used as a register memory address area, and the upper-layer application accesses BAR2. To read such resources, the device file must be opened. Before using I/O resources, the driver must apply for the right to use. It should be emphasized here that the relevant I/O memory resources must be mapped; otherwise, they cannot be used by upper-layer applications. Through the production of the user login interface, virtual experiment main interface and three-dimensional UI interface, the software part of the virtual assembly system designed in this paper has been realized. After functional testing, the produced system software can be released through Unity3d software.

6. Conclusion

This paper proposes a cloud data integrity verification algorithm for intelligent accounting informatization based on cloud computing, aiming to solve accounting data's security and integrity problems in the cloud environment. By introducing digital signature, hash algorithm and blockchain technology, combined with the advantages of a cloud computing platform, this algorithm realizes the integrity verification of accounting data in the process of storage, transmission and processing. It can effectively prevent security risks such as data tampering, loss and illegal access. Through experimental verification, the algorithm proposed in this paper shows excellent performance in large-scale data verification tasks, with high verification efficiency and accuracy, especially when dealing with real-time data, and can effectively meet the high standard of data security requirements of intelligent accounting systems.

In addition, this paper also considers the data dynamics and scalability in the cloud computing environment and designs a verification algorithm with good adaptability, which can operate stably in complex and changing cloud environments and adapt to enterprises of different sizes and their accounting informatization needs. Compared with the traditional data integrity verification methods, this algorithm not only improves the security of the system but also improves the verification efficiency, reduces the consumption of system resources, and provides a practical solution for the realization of an intelligent accounting informatization system.

However, although this study has achieved better results in algorithm design and experimental verification, there is still room for further optimization. Future research can explore in depth the fault tolerance, real-time, and compatibility of the algorithms, especially how to maintain their efficiency and accuracy in the face of large data volumes, high-frequency accesses, and complex operations. In addition, the algorithms proposed in this study can be combined with other intelligent technologies to further enhance the intelligence level and security of accounting informatization systems.

Declaration

Data Availability: The experimental data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest: The authors declared that they have no conflicts of interest regarding this work.

Funding: This work was supported by Sichuan University Jinjiang College 2024 Campus level Applied Brand Course Construction Project Intelligent Accounting (Project Number: 7)

Authors' Contributions: Jun Zhang is responsible for designing the framework, analyzing the performance, validating the results, and writing the article. Liang Xian is responsible for collecting the information required for the framework, provision of software, critical review, and administering the process.

References

[1] Zhang C, Xie T, Yang K, et al. Positioning optimisation based on particle quality prediction in wireless sensor networks. IET Networks. 2019;8(2):107-113.

[2] Xu W. Toward human-centered AI: a perspective from human-computer interaction. interactions. 2019;26(4):42-46.

[3] Diederich S, Brendel AB, Morana S, Kolbe L. On the design of and interaction with conversational agents: An organizing and assessing review of human-computer interaction research. Journal of the Association for Information Systems. 2022;23(1):96-138.

[4] Witanto, E. N., Stanley, B., & Lee, S. G. (2023). Distributed data integrity verification scheme in multi-cloud environment. *Sensors*, *23*(3), 1623.

[5] Zhang C, Roh BH, Shan G. Poster: Dynamic clustered federated framework for multi-domain network anomaly detection. Companion of the 19th International Conference on Emerging Networking EXperiments and Technologies. 2023 Dec:71-72.

[6] Shneiderman B. Human-centered artificial intelligence: Three fresh ideas. AIS Transactions on Human-Computer Interaction. 2020;12(3):109-124.

[7] Ortner D. Creative arts as self-care: vicarious trauma, resilience and the trainee art therapist. International Journal of Art Therapy. 2024;29(1):1-1.

[8] Kehr T, Haeyen S. Drawing your way out: Imagery rehearsal based art therapy (IR-AT) for post-traumatic nightmares in borderline personality disorder. Journal of Clinical Psychology. 2024;80(5):1015-1030.

[9] Estrada Gonzalez V, Meletaki V, Walker M, et al. Art therapy masks reflect emotional changes in military personnel with PTSS. Scientific Reports. 2024;14(1):7192.

[10] Malhotra B, Jones LC, Spooner H, et al. A conceptual framework for a neurophysiological basis of art therapy for PTSD. Frontiers in Human Neuroscience. 2024;18:1351757.

[11] Ong EK, Tan UTE, Chiam M, Sim WS. The employment of art therapy to develop empathy and foster wellbeing for junior doctors in a palliative medicine rotation–a qualitative exploratory study on acceptability. BMC Palliative Care. 2024;23(1):84.

[12] Barnett KS, Vasiu F. How the arts heal: a review of the neural mechanisms behind the therapeutic effects of creative arts on mental and physical health. Frontiers in Behavioral Neuroscience. 2024;18:1422361.

[13] Nandagopal, M., Manavalan, T., Kumar, K. P., Manogaran, N., Kesavan, D., Kumar, G., & Al-Khasawneh, M. A. (2025). Enhancing energy efficiency in cloud computing through task scheduling with hybrid cuckoo search and transformer models. *Discover Computing*, *28*(1), 199.

[14] Isis P, Bokoch R, Fowler G, Hass-Cohen N. Efficacy of a single session mindfulness-based art therapy doodle intervention. Art Therapy. 2024;41(1):11-20.
[15] Weinfeld-Yehoudayan A, Czamanski-Cohen J, Cohen M, Weihs KL. A theoretical model of emotional processing in visual artmaking and art therapy. The Arts in Psychotherapy. 2024;90:102196.
[16] Frankel AV, Stern EM, Fritsch K, Berkhout S. Surface and depth in sensory ethnography: casting bodied experience in an arts-based interviewing practice. International Journal of Qualitative Methods. 2024;23:16094069241270460.
[17] Geraghty KA, Vahabzadeh LJ, Simonet R. Art therapy in a multidisciplinary team with young men in prison. International Journal of Art Therapy. 2024;29(1):39-44.
[18] Oepen R, Gruber H. Art-based interventions and art therapy to promote the health of migrant populations–a systematic literature review of current research. Arts & Health. 2024;16(3):266-284.
[19] Xia Y, Deng Y, Tao X, Zhang S, Wang C. Digital art exhibitions and psychological well-being in Chinese Generation Z: An analysis based on the SOR framework. Humanities and Social Sciences Communications. 2024;11(1):1-15.
[20] Nippatla, R. P., & Palanisamy, P. (2020). Optimized cloud architecture for scalable and secure accounting systems in the digital era. *International Journal of Multidisciplinary and Current Research*, *8*.
[21] Vaisvaser S, King JL, Orkibi H, Aleem H. Neurodynamics of relational aesthetic engagement in creative arts therapies. Review of General Psychology. 2024;28(3):203-218.
[22] West HW, Dangas K, Antoniades C. Advances in clinical imaging of vascular inflammation: a state-of-the-art review. Basic to Translational Science. 2024;9(5):710-732.
[23] Iyendo TO, Uwajeh PC, Oseke BN, et al. Effect of Audio-Visual-Based Art and Music Therapy in Reducing Adolescents Post Traumatic Stress Disorder. Journal of Adolescent Health. 2024;74(2):283-291.
[24] García-Figueiras R, Baleato-González S, Luna A, et al. How imaging advances are defining the future of precision radiation therapy. Radiographics. 2024;44(2):e230152.
[25] Endacott JL, Warren J, Hackett-Hill K, Lalonde A. Arts integrated historical empathy: Preservice teachers' engagement with pluralistic lived experiences and efforts toward instructional application. Theory & Research in Social Education. 2024;52(3):414-457.
[26] Bhurani, K., Dogra, A., Agarwal, P., Shrivastava, P., Singh, T. P., & Bhandwal, M. (2024). Smart Contracts for Ensuring Data Integrity in Cloud Storage with Blockchain. *EAI Endorsed Transactions on Scalable Information Systems*, *11*(6).