Journal of Sustainable Solutions
ISSN : 3048-6947 | Vol. 1 | Issue 4 | Oct - Dec 2024 | Peer Reviewed & Refereed

Automated Data Integrity Checks for Financial Software Systems

Chinmay Mukeshbhai Gangani
Independent Researcher, USA.

Accepted: 01/10/2024 Published: 02/11/2024 * Corresponding author

How to Cite this Article:

Gangani, **C** (2024). Automated Data Integrity Checks for Financial Software Systems, *Journal of Sustainable Solutions*, 1(4), 197-207.

DOI: https://doi.org/10.36676/j.sust.sol.v1.i4.52

Abstract

The integrity and security of the data that is outsourced are often threatened by an untrusted cloud server, despite the fact that cloud storage offers simple data outsourcing options. Designing security techniques that enable users to verify data integrity with reasonable computing and communication overheads is thus of utmost importance. The goal of this study is to create AI Data Quality Co-pilots, which are advanced systems designed to automatically assess and improve data quality in real time. According to AI Data Quality Co-pilots, future concerns like data drift, privacy, and inclusion won't affect the AI model's dependability or impartiality. Additionally, it discusses how co-pilot applications increase real-time base-level decision-making content and decrease erroneous fraud signals, as well as how ethical AI may be achieved by detecting and correcting biases. Businesses may expand and enhance AI with appropriate data management and maintain effective AI models with consistent high-quality data inputs by linking these co-pilots. Cloud computing is a popular option for many firms since it offers lower-cost computer service outsourcing. Since data owners must depend on outside cloud storage providers to manage their data, data integrity is a crucial concern in cloud computing. In order to improve security and guarantee the quality of data that is outsourced, researchers have been creating new algorithms for data integrity techniques in cloud storage. This article discusses the stages, traits, and categorisation of data integrity solutions in addition to highlighting security concerns and potential assaults on cloud storage. Additionally included is a comparative study of various tactics in relation to cloud storage. Additionally, taking into account the intended design objectives, the overhead characteristics of auditing system models in cloud computing are investigated. Organisations may make well-informed judgements about their cloud storage solutions by comprehending and resolving these aspects, taking performance and security into account.

Keywords: - Cloud Storage, Ethical AI, Cloud Computing, Computing Services, AI Models, Data Integrity, Comparative Analysis, Model's Reliability, Data Management, Classification, Co-Pilot’s, Data Drift, Privacy, Security and Performance.

I. INTRODUCTION

Cloud storage has gained popularity as a new computer paradigm because it offers excellent data management and storage capabilities. Pay-as-you-go cloud data outsourcing is becoming more and more popular among businesses and consumers [1, 2]. However, cloud customers no longer have to worry about local storage thanks to cloud outsourced storage services. Verifying data integrity becomes a crucial security issue for the use of cloud technologies. On the one hand, there will be a significant increase in communication and compute overhead if the whole data is downloaded often for integrity verification [2, 3]. In contrast, the cloud service provider (CSP) may decide to hide data loss or corruption in order to preserve user confidence if the cloud storage devices are broken or if hackers steal the data that was outsourced. Additionally, the CSP may purposefully remove data that isn't used

as often in order to save storage space or disclose users' private information for curiosity. As a result, cloud users need to figure out how to effectively check the accuracy of the data that is being outsourced [3, 4].

To address the aforementioned issue, the techniques described in the literature provide models that enable both private and public verification. Private verification only enables users to independently verify the integrity of the data that has been outsourced, which places a significant computational load on users with limited resources. Conversely, public verification lowers the computational cost for the user by adding a third-party authority (TPA) to confirm integrity [4]. Public verification has drawn increased attention because of its potential for real-world use. The RSA-based homomorphic linear authenticator is used in the first publicly verified cloud data integrity system to verify the accuracy of data that is outsourced. a publicly accessible data integrity verification system signed by Boneh–Lynn–Shacham (BLS). Multiple tags are combined into a single tag in this technique [4, 5], and the tag is validated using bilinear maps. These systems, however, come with high computational and communication costs for costly exponentiation processes. A public verification approach based on algebraic signatures that achieved effective integrity verification without comparing with original data by using a short bit string compressed by a data block. Regretfully, it is not safe against the replay attack [5, 6]. Moreover, dynamic data updating is not supported.

Researchers have proposed a number of methods to update data without download the whole set in order to provide dynamic updating. The first verification approach uses a rank-based authorised skip list to facilitate dynamic data updating. To perform integrity verification, however, the list needs a great deal of auxiliary data [5, 6]. High computational and communication expenses result from it. A variety of enhanced strategies are then put out, including. It should be noted that in order to facilitate dynamic data updating, the Divide and Conquer Table (DCT) data structure was established. The table's separation into distinct parts indicates that DCT outperforms Index-Hash-Table (IHT) in terms of data block insertion and deletion. But there are still some issues.

AI co-pilots are AI helpers or tools designed to help people do activities more quickly. They are often used in applications to provide context-sensitive help, automation, or suggestions depending on inputs [5, 6]. To interact with people, these co-pilots use advanced AI models, such as machine learning and natural language processing (NLP). Among the most well-known are Microsoft Co-pilot for text in programs like Word or Excel for activities pertaining to document editing, data analysis, and report compilation, and GitHub Co-pilot for programming, which provides code recommendations. Stated differently, an AI Co-pilot collaborates on a project to boost productivity and make more sensible decisions [6, 7].

Furthermore, via sophisticated analysis, automation, and decision-making assistance, AI is progressively emerging as a major solution in businesses and organisations. Nonetheless, there is a strong inverse correlation between the quality of the data used by AI systems and that data. Making decisions about data timeliness, correctness, and dependability in the context of large data is challenging nowadays. Furthermore, [8, 9], AI models built with poor quality, primarily biased, or insufficient data may result in distortions, which will further perpetuate these errors and injustices against certain individuals. As artificial intelligence (AI) grows and these intelligent technologies are increasingly used in basic applications, quality data standards are becoming more than just a technical need [8]—they are also morally required. The AI Data Quality Co-pilots are described in this study as intelligent systems designed to address novel data problems that emerge in the setting of AI. The majority of co-pilots optimise the data supplied to AI models in dynamic scenarios by continuously validating and evaluating the input.

The process of methodically identifying, analysing, and assessing risks that might result from financial transactions, operational operations, regulatory changes, or external market circumstances is known as risk assessment in SAP systems [8, 9]. In order to enable well-informed risk management strategies, it seeks to assess the probability and possible effect of these risks on organisational goals [10, 11]. Automated risk assessment makes use of new technology, especially algorithms that utilise machine learning, to improve and expedite this procedure. Real-time analysis of massive amounts of data from SAP's integrated accounting modules enables automated systems to spot trends, abnormalities, and possible hazards more quickly and precisely than conventional techniques.

In order for integrity property standardisation to progress, requirements and implementation specifications have not been articulated coherently. Now, a lot of work has to be done to standardise the integrity attributes of systems. A starting point is given in this study [11,12]. An analysis of the computer security needs for military tactical and embedded computer systems served as the initial catalyst for this research, highlighting the necessity of integrity standards for military systems. Integrity concerns have risen in significance as the military has become more reliant on intricate, globally networked computer systems. The hazards of disclosing information, especially volatile information that must be utilised as once after it is released, are often minimal [12].

In many systems, such as those that deal with school data, travel reservations, medical information, finances, insurance, and people, integrity may be considered more significant than secrecy. While maintaining the security of data in these kinds of systems is often necessary [11,12], it is also vital that data not be altered or tampered with in an unauthorised manner. Computer systems that are embedded are also included in this system classification. These systems are parts that are integrated into a bigger system to carry out one or more specialised (often control) tasks [13]. Since they often lack a human interface to help ensure proper system performance, they provide a more distinctive perspective on the significance of integrity. Military weaponry systems are not the only use for embedded computer systems [14]. Commercial examples include robotic actuator control systems, automated milling machines, radiological imaging equipment, anti-lock brake systems, and aviation avionics. Integrity may be seen in the historical perspective of computer system protection mechanism development as well as in the viewpoint of relative value. The initial purpose of many protection measures was to maintain integrity [14].

They weren't acknowledged as being as crucial to maintaining secrecy until much later. The possibility that programs may access memory—either main memory or auxiliary memory, such disks—that was not assigned to them was among the first worries [15]. Systems have to safeguard the resources allotted to the concurrent execution of routines from unintentionally altering each other as soon as they started allocating resources to many programs simultaneously (e.g., multitasking, multiprogramming, which and time-sharing) [14]. In addition to interrupt, privilege, and safeguarded address spaces implemented in hardware and software, this increased system concurrency resulted in a type of interleaved sharing of the processors using two or more processor states (for example, one for problem or user state and a second for control or system state).

1.1 Data Integrity

When most people think of integrity in computer systems, they think of data integrity first. It indicates qualities of data including quality, correctness, authenticity, timeliness, accuracy, and precision, according to several [15]. The preservation of information's meaning, the comprehensiveness and consistency of its representation within the framework of the system, and its connection to representations outside the system are all aspects of data integrity [14]. It entails the effective and accurate functioning of computer hardware and software with regard to data, as well as, where appropriate, [16] the proper actions of computing system users, such as data input. When AISs handle

multiple different types of data with the same equipment or share multiple different user groups, data integrity is the main problem. Because of the variety and interaction of information that these systems frequently deal with, as well as the potentially large and widely used number of users and system nodes that must interact via these systems, it is a concern in large-scale, distributed, and networked processing systems [16, 17].

1.2 Systems Integrity

Here, "systems integrity" refers to the efficient and accurate use of computational resources. Although the idea of systems integrity is broad for computer systems, it has particular ramifications for embedded systems whose control depends on system sensors. Fault tolerance and systems integrity are strongly connected [15, 16]. Because it requires fault tolerance, a component of computing that is sometimes misunderstood to be limited to the hardware level, this part of integrity is frequently left out of the typical discussions of integrity. The embedded system only has less user-provided fault tolerance; systems integrity is just a hardware problem on the surface and applies equally to the AIS context. It is also strongly connected to the topic of system safety in this context [16], such as the safe functioning of an aeroplane that uses embedded computers to maintain steady flying [16, 18].

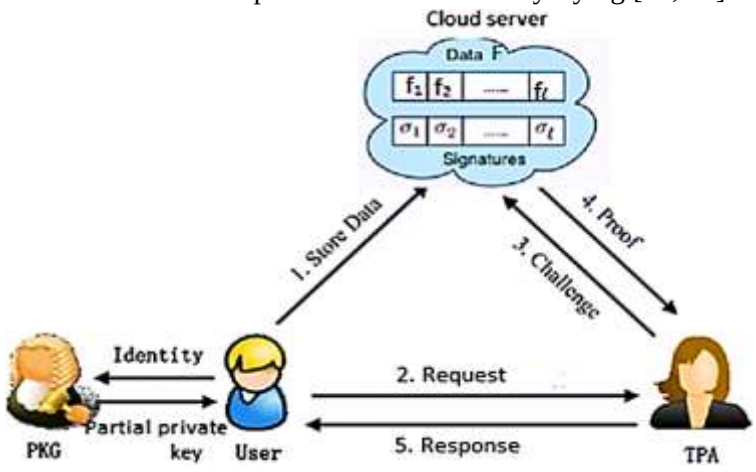

Fig. 1 Data integrity. [18]

1.3 Integrity Goals

We reframe the general integrity objective into the following particular goals in what we perceive to be the order of increasing difficulty to accomplish, using the integrity goal previously indicated and the expansions we suggest [18, 19]. All of these objectives may need additional risk reduction strategies, and some may react to mechanisms recognised to provide a certain level of confidence [19, 20].

1.4 Preventing Unauthorized Users

- **From Making Modifications:** Both data and system resources are addressed by this objective. Inappropriate access to the system, its resources, and its data is considered unauthorised usage. Changes to the system [11], its resources, and the user or system data that was initially stored, including the addition or deletion of such data, are all considered unauthorised modifications. This objective is the reverse of the confidentiality requirement with regard to user data [19,20]: integrity restricts the flow of information into the stored data, while confidentiality restricts the flow of information out of the stored data.
- **Maintaining Internal and External Consistency:** This objective covers both systems and data. It discusses how interdependent data may be consistent with themselves as well as how data can be consistent with the real-world environment it reflects [20, 22]. Maintaining internal consistency becomes more difficult in distributed computing systems with replicated and dispersed data.

– **Preventing Authorized Users:** From Making Inappropriate Changes Integrity's last objective is the most nebulous and often entails risk-reduction techniques or protocols as opposed to strict system inspections [21, 22]. For instance, an employee may be permitted to move money to certain corporate accounts, but they should not conduct random or fraudulent transfers in order to prevent inappropriate adjustments [22].

II. CLOUD COMPUTING'S APPEAL LIES IN ITS DYNAMIC AND FLEXIBLE

Negotiable services based on Service Level Agreements (SLAs) provide consumers access to almost infinite computer resources [22, 23]. The National Institute of Standards and Technology (NIST) claims that cloud computing provides a pay-per-use model that can be quickly provisioned, allowing for on-demand, readily available, and configurable network access to resources from a shared pool with less interaction from service providers and less management work [22].

Even if there are many different data services available, data owners are reluctant to trust cloud service providers (CSPs) with their sensitive data for third-party cloud storage because of worries about the CSPs' integrity [22, 23] and the collaborative character of cloud storing settings. Data storage and computation are the main components of cloud computing, and cloud storage and Infrastructure as a Service (IaaS) are closely related. Cloud customers that utilise IaaS often don't have insight into the exact location of their outsourcing data in the cloud storage and the computers that are handling processing activities [23]. Because of this, protecting data privacy in cloud storage is a major security concern that is made worse by malevolent users, which leads to problems with data integrity and confidentiality. This is a serious security risk for cloud storage, since the achievement of cloud computing depends on people having faith in remote cloud storage of their information.

In order to preserve the data integrity of business organisations' information, Google Cloud recently launched Zebra technologies, which are based on a security command central (SCC) and security operation centre (SOC). These technologies are designed to identify detrimental threats like crypto mining activity, data exfiltration, possible malware diseases, brute force SSH attacks, etc. Many cloud data integrity systems and survey studies have surfaced in recent years; however, they are too narrowly focused to fully solve particular data integrity issues. diverse data integrity methods and validation types for cloud storage, data audits from single copies to multiple replicas, Proof of Retrievability, and diverse data integrity protocols are a few of these studies. Nevertheless, these surveys often fail to provide a thorough grasp of data integrity tactics and how they are categorised [21].

Table 1 Possible Attack and Threat Types at the Storage Level Data Integrity with Countermeasures. [21, 23]

Potential Attacks	Storage Issues	Threats	Mitigation Solution with references	Applied Methods
DoS	Lack of a forecast format to estimate the time and storage needed to process and store data in the cloud, as well as the risk to data.	Vulnerable service is used in lieu of the original service [22].	Suggested mechanism for authorisation and authentication. The suggested signature-based method offered a means of detecting and preventing incursions [23].	Attribute-Based Proxy Signature in the Kerberos Protocol: Enhanced Dynamic Immune Algorithm (IDIA)

Phishing	Unauthorised access to physical cloud storage and a lack of storage monitoring.	Disclosure of data confidentially	Provide a method for phishing detection.	Hyper-parameter classifier tweaking and a hybrid classifier technique [19].
Brute Force Attack / online dictionary Attack	Access to physical cloud storage without accreditation.	Violation of data authenticity and disclosure of data confidentiality [24].	Provide a plan for data obfuscation.	Replacement technique using the Least Significant Bit (LSB).
MITC Attack	Inadequate defence against malevolent external and internal threats.	Unusual availability of services [30].	Provide a method for string authentication.	Fuzzy extractors and chaotic maps [20].
Risk Spoofing	Data lock-in, unreliable cloud storage, and CSP monitoring shortcomings.	Internal security is lacking, and there are recording violations.	keeping an eye on safe data rules.	Attribute-based encryption (ABE) or symmetric searchable encryption (SSE).
Data Loss/Leakage Propose	CSP's incapacity, ongoing storage monitoring, and lack of scalability [30].	Malicious Cloud Computing Provider, Malicious Insider [11].	Method of data integrity.	Techniques for public data audits and data encryption.
Identity Theft	Not accredited. data danger, unreliable cloud storage, and physical cloud storage access [20].	Violating SLAs and security rules [30].	privacy beach protection and a password-based login system [26].	Compact password-authenticated exchange of keys protocol (Compact AKE), key-based semantically secure Blooming filter (KSSBF), OTP, and Evolutionary System Model-based privacy preservation (EMPPC) [23].

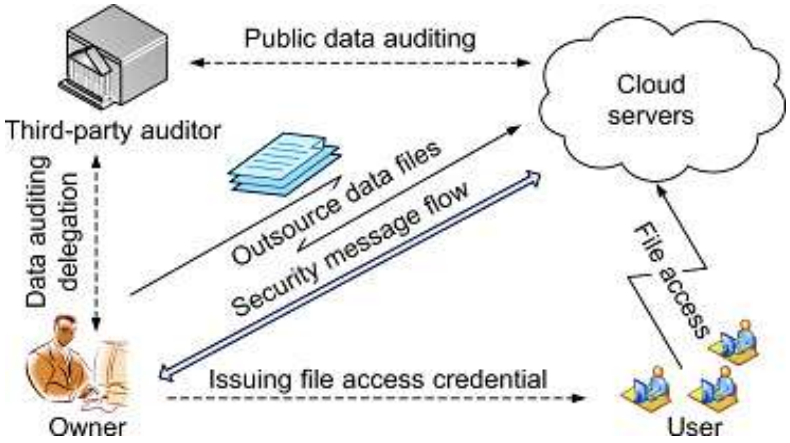

Fig. 2 Complete Data Integrity Technique Cycle. [26]

Table 2 Data Integrity Scheme Phases That Are Classified. [11]

Ref.	Technical Methods	Data Processing Phase			Phase of Acknowledgements	Auditing Phase			
		Initial Phase	Phase of Key and Signature Generation	Encryption		Using TPA	Using the Client/ Data Owner	Challenge phase	Stage of Proof verification
[1]	Maintaining the integrity checking model and guaranteeing confidentiality.	Yes	Yes	No	No	Yes	Yes	Yes	Yes
[5]	Data integrity and privacy are mitigated by data auditing.	Yes	Yes	Yes	No	No	No	No	No
[22]	Data sharing and identity-based integrity auditing.	No	No	Yes	No	Yes	No	Yes	Yes

[28]	Data sharing and identity-based integrity auditing.	Yes	Yes	Yes	No	Yes	Yes	No	No
[30]	Algebraic Signatures-Based Data Integrity Auditing.	Yes	No	Yes	Yes	Yes	Yes	Yes	No

III. CHARACTERISTICS OF DATA INTEGRITY TECHNIQUE

This review article focusses on a number of data integrity quality aspects that are individually crucial to cloud storage security [14, 25]. These are:

- **Public Auditability:** At the request of data owners, the auditability scheme verifies the accuracy of data that is outsourced from them and stored in cloud storage by TPA [25, 26].
- **Audit correctness:** Only when both CSP and TPA are acting honourably and CSP, the data owner, correctly adheres to the established procedure for data storage will the proof message of CSP be able to pass the TPA validation test [24, 26].
- **Auditing soundness:** CSP must keep all of the data owner's outsourced data in cloud storage in order to pass TPA's verification test [26, 27].
- **Error localization at block level:** During the verification process, it is useful to identify incorrect blocks in a cloud storage file.
- **Data Correctness:** Using the information from the available replica block in cloud storage, it assists in fixing faulty data blocks [27, 28].
- **Stateless Auditor:** A stateless auditor is not required during verification in order to preserve, archive, or update prior verification findings for use in the future.
- **Storage Correctness:** Even if the data is partly tempered or lost, CSP creates a report that demonstrates that all data is fully saved in cloud storage. Consequently, the system must ensure that data owners' outsourced data matches their previously stored data [28, 29].
- **Robustness:** When using a probabilistic data integrity technique, mistakes in smaller data sets should be found and fixed.

IV. CHALLENGES OF DATA INTEGRITY TECHNIQUE IN CLOUD ENVIRONMENT

There are usually some basic problems about the security issues with data integrity techniques in cloud computing:

A. Risk to integrity of data:

– Cloud services are hindered by many harmful assaults throughout the worldwide acquisition period if database, network, etc. integrity is not well maintained [30].
– Issues with data availability and integrity arise when CSP makes unauthorised modifications to the data.
– Another issue with data integrity is data segregation across cloud users in cloud storage.

As a result, data integrity techniques must include SLA-based patch management policies, standard validation methods against unauthorised usage, and sufficient security settings [23, 30].

B. Dishonest TPA:

– TPA may damage CSP's reputation by producing incorrect integrity verification notifications.
– Through regular verification interaction messages with cloud storage, bad attackers might assist TPA in exploiting sensitive data [29].

C. Dishonest CSP:

i) Via frequent verification interaction messages with cloud storage, malicious attackers might help TPA take use of private information [29].

ii) CSP has the authority to alter a file's real content and use it for their own purposes. However, in both situations, the data owner is unable to identify the real offender.

D. Forgery Attack at Cloud Storage:

– A proof message issued by CSP for the blocks identified by the challenge message to react TPA may be forged by an outside attacker [10, 30].
– It is possible for malicious auditors to fabricate an audit evidence that passes the data integrity check.

E. Data modification by an insider malicious user into cloud storage:

Even if an insider hostile user modifies the network channel's interaction messages, they may still subvert or change a data block at their discretion and deceive the auditor and data owner into believing that the data blocks are securely stored in the cloud [30].

V. CONCLUSION

It is now clear that the quality of the data utilised throughout an AI system's life cycle strongly affects the system's performance in today's comprehensive and data-driven world. In addition to creating significant operational, financial, and reputational costs for businesses, low data quality may lead to subpar, ineffective, and even immoral performance and erode public trust in AI systems. Thus, acquiring high-quality data that will be used in the creation of appropriate, effective, and moral artificial intelligence systems becomes crucial. Co-pilots for AI data quality are crucial in addressing the wide range of data quality problems that are addressed in the area of AI.

It is uncomfortable to ensure data owners that the intact condition of outsourced data in cloud storage settings has become a major security concern due to the steadily growing popularity of appealing and cost-effective cloud services. We have made an effort to draw attention to a number of problems with cloud data integrity and the related techniques to solving them, which will provide researchers a clear picture and guidance. The state-of-the-art in the aforementioned research field will lead to further breakthroughs in a number of domains, such as controlling social media fat-forms, secure financial services, and cloud-based sensitive health care. The phases of data integrity, the features of data integrity schemes, the classification of data integrity strategies according to proposal types, the types of data and verification schemes, and the intended design challenges of data integrity strategies based on performance overhead have all been covered in this paper.

VI. REFERENCES

[1] Schultz, Pim. Using machine-learning models for operational exception handling: a case study at IBM. MS thesis. University of Twente, 2017.

[2] Singh, Kishore, and Peter J. Best. "Design and implementation of continuous monitoring and auditing in SAP enterprise resource planning." International Journal of Auditing 19.3 (2015): 307-317.

[3] Singh, Kishore, Peter Best, and Joseph Mula. "Automating vendor fraud detection in enterprise systems." Journal of Digital Forensics, Security and Law 8.2 (2013): 1.

[4] Niesen, Tim, et al. "Towards an integrative big data analysis framework for data-driven risk management in industry 4.0." 2016 49th Hawaii international conference on system sciences (HICSS). IEEE, 2016.

[5] Büsch, Sebastian, Volker Nissen, and Arndt Wünscher. "Automatic classification of data-warehouse-data for information lifecycle management using machine learning techniques." Information Systems Frontiers 19 (2017): 1085-1099.

[6] Sinha, Akinchan Buddhodev. "Role of Information Technology in Business Risk Management." IUP Journal of Systems Management 9.4 (2011).

[7] Emmenegger, Sandro, et al. "Towards a procedure for assessing supply chain risks using semantic technologies." Knowledge Discovery, Knowledge Engineering and Knowledge Management: 4th International Joint Conference, IC3K 2012, Barcelona, Spain, October 4-7, 2012, Revised Selected Papers 4. Springer Berlin Heidelberg, 2013.

[8] Husebø, Ivan-Louis Miranda, and Andreas Kvist. Decreasing Manual Workload by Automating SAP Travel Expense Workflows. MS thesis. University of Stavanger, Norway, 2018.

[9] S. K. Rachakatla, P. Ravichandran, and J. R. Machireddy, "Scalable Machine Learning Workflows in Data Warehousing: Automating Model Training and Deployment with AI," Australian Journal of Machine Learning Research & Applications, vol. 2, no. 2, pp. 262-286, 2022.

[10] C. Bird, D. Ford, T. Zimmermann, N. Forsgren, E. Kalliamvakou, T. Lowdermilk, et al., "Taking Flight with Copilot: Early insights and opportunities of AI-powered pair-programming tools," Queue, vol. 20, no. 6, pp. 35-57, 2022.

[11] A. M. Dakhel, V. Majdinasab, A. Nikanjam, F. Khomh, M. C. Desmarais, and Z. M. Jiang, "GitHub copilot AI pair programmer: Asset or liability?" Journal of Systems and Software, vol. 203, p. 111734, 2023.

[12] Nica and V. Stehel, "Internet of things sensing networks, artificial intelligence-based decision-making algorithms, and real-time process monitoring in sustainable industry 4.0," Journal of Self-Governance and Management Economics, vol. 9, no. 3, pp. 35-47, 2021.

[13] Y. Bao, G. Hilary, and B. Ke, "Artificial intelligence and fraud detection," Innovative Technology at the Interface of Finance and Operations: Volume I, pp. 223-247, 2022.

[14] R. Mohan, M. Boopathi, P. Ranjan, M. Najana, P. K. Chaudhary, and A. K. Chotrani, "AI in Fraud Detection: Evaluating the Efficacy of Artificial Intelligence in Preventing Financial Misconduct," Journal of Electrical Systems, vol. 20, no. 3s, pp. 1332-1338, 2024.

[15] Shen J, Liu D, He D, Huang X, Xiang Y (2017) Algebraic signatures-based data integrity auditing for efcient data dynamics in cloud computing. IEEE Trans Sustain Comput 5(2):161–173 95.

[16] Wang B, Li H, Liu X, Li F, Li X (2014) Efcient public verifcation on the integrity of multi-owner data in the cloud. J Commun Netw 16(6):592–599.

[17] Yu Y, Li Y, Yang B, Susilo W, Yang G, Bai J (2017) Attribute-based cloud data integrity auditing for secure outsourced storage. IEEE Trans Emerg Top Comput 8(2):377–390 97. Zhu H, Yuan Y, Chen Y, Zha Y, Xi W, Jia B, Xin Y (2019) A secure and efcient data integrity verifcation scheme for cloud-iot based on short signature. IEEE Access 7:90036–90044 98.

[18] Wang H, He D, Tang S (2016) Identity-based proxy-oriented data uploading and remote data integrity checking in public cloud. IEEE Trans Inf Forensic Secur 11(6):1165–1176 99.

[19] Thakur AS, Gupta P (2014) Framework to improve data integrity in multi cloud environment 100.

[20] Zhang C, Xu Y, Hu Y, Wu J, Ren J, Zhang Y (2021) A blockchain-based multi-cloud storage data auditing scheme to locate faults. IEEE Trans Cloud Comput 10(4):2252–2263.

[21] Subha T, Jayashri S (2014) Data integrity verifcation in hybrid cloud using ttpa. In: Networks and communications (NetCom2013). Springer, pp 149–159.

[22] Mao J, Zhang Y, Li P, Li T, Wu Q, Liu J (2017) A position-aware merkle tree for dynamic cloud data integrity verifcation. Soft Comput 21(8):2151–2164.

[23] Han S, Liu S, Chen K, Gu D (2014) Proofs of retrievability based on mrd codes. In: International Conference on Information Security Practice and Experience. Springer, pp 330–345.

[24] Kaaniche N, El Moustaine E, Laurent M (2014) A novel zero-knowledge scheme for proof of data possession in cloud storage applications. In: 2014 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing. IEEE, pp 522–531.

[25] Khedr WI, Khater HM, Mohamed ER (2019) Cryptographic accumulatorbased scheme for critical data integrity verifcation in cloud storage. IEEE Access 7:65635–65651.

[26] Khatri TS, Jethava G (2013) Improving dynamic data integrity verifcation in cloud computing. In: 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT). IEEE, pp 1–6.