

Secure Dataset Verification Using Blockchain and Merkle Trees

D. John Subuddhi¹, S. Keerthi Srivalli², Yesaswini Swarna³, S. Karthik⁴, P. Chetan Satish⁵

¹²³⁴⁵Department of Computer Science and Engineering, Vishnu Institute of Technology, Bhimavaram, India

Abstract - Most of the models rely heavily on data. That data may be large, continuously updating. Existing data management models lack many aspects like trustworthy of data and which makes them weak to attacks. Block chain is current demanding technology for increasing the assurance of data integrity in distributed systems. Still using cryptographic hash methods results in ineffective results. So, in this paper we introduce a model which uses hash generation to create a merkle root node and by comparing this merkle root node with later produced root node value, we can confirm whether a dataset has been modified or not. The current existing models may have this creation of merkle tree but no model has other features like calculation of trust score and tampering alert system. So, we integrated other features like developing dynamic trust score, which automatically reduces if there is detection of tampering and role based access of data to control other unauthorized actions and differential dataset backup which helps to restore the original dataset and smart alert system which sends alerts immediately when tamper is detected. By combining all these features into a single framework, it introduces new capabilities such as transparent tracking of data, automated trust evaluation and dataset versioning that were not introduced in previous models. The primary beneficiaries include ML practitioners, data researchers, and organizations.

Key Words: Block chain-based Security, Data set Integrity, SHA256 Hashing, Binary Markel Tree, Smart Contracts, Role-Based Access Control(RBAC).

1. INTRODUCTION

This study says that distributed and data-driven systems [1] are growing fast. In these systems it is very important to make sure that the data is correct and trustworthy. Many modern applications use intelligence, cyber security [2] and other platforms that rely heavily on shared datasets that are always being updated and have a lot of data [3]. This makes it more likely that someone could change the data without permission tamper with it or poison it which can hurt the systems reliability. So we need a way to always check that the data is correct and trustworthy [4].

The solutions we have now are mostly. Depend on trusted authorities, which makes them weak to failures and attacks. Also most systems do not have a way to check data integrity in time evaluate trust and send out alerts automatically [5]. Not having a framework that supports

checking integrity tracking where the data comes from and evaluating trust is a gap in research that this study is trying to fill. This study tested the idea that it can help make digital data systems more trustworthy [6].

Making sure the data is correct directly affects how reliable the final results, machine learning models and decision-making processes are [7]. By providing a way to prevent tampering this study helps improve accountability and reduce security risks [8], which's important for applications that rely on data. We have already used hash functions [9] to detect when data has been changed but they are not good enough. Merkle Trees [10] are a way to efficiently check datasets. With Merkle Trees some records from a dataset are combined to get a hash value, and then all these hash values are combined to get a single Merkle root value [11].

The big problem this research found is that there is no way to guarantee that the data is correct and trustworthy in a distributed environment [12]. The main goal of this study is to design a framework that uses blockchain and combines Merkle Tree- based verification [13] with a trust scoring mechanism, time tamper detection [14] and role-based access control [15]. What makes this approach special is that it creates Merkle Trees and different versions of datasets. The solution we propose shows that using blockchain technology with Merkle Tree-based verification [16] can greatly increase the assurance of data integrity in distributed systems. We expect the system to be able to detect tampering in time to update trust scores based on verification results to send out alerts when tampering occurs and control access based on roles. This is possible because of hashing, the efficiency of Merkle Trees, and the decentralized trust model that blockchain technology offers. We chose to use blockchain technology because it does not rely on authorities and Merkle Trees because they make it efficient to verify large datasets without needing to access the whole dataset.

This makes the model very suitable for distributed environments where data is frequently updated and shared among parties that do not trust each other. The advantages of this proposed system are that it provides guarantees of integrity, non-repudiation and tamper resistance. It also reduces storage requirements by storing hash values on the blockchain supports tampering verification enables proper auditing and allows real-time integrity monitoring making it feasible, for real-world use.

2. LITERATURE REVIEW

The available literature on cryptographic data integrity and security based on blockchain [17] provided a solid theoretical background but leaves a number of practical issues unaddressed. Initial ideas by Merkle, the merkle tree idea of hash generation to perform efficient integrity tests, were subsequently used as a component in data secure systems [18]. The use of Bitcoin to popularize blockchain technology showed how the cryptographic hashing can be used to provide immutability and trustiness [19].

Other platforms related to this notion include Ethereum which added extra functionality by incorporating things like smart contracts which further allowed programmable and automated verification of things [20]. A number of studies have been conducted on blockchain as a data security and provenance tool. The concept of decentralized privacy-sensitive data management based on the use of blockchain was suggested by Zyskind et al. and outlines its capability to safeguard sensitive data [21]. Hasan and Salah proposed blockchain provenance that ensures data integrity and traceability [22].

Secure data and integrity verification of distributed systems may be implemented using blockchain and Merkle Trees [23]. Although these methods are used to determine tampering, they have much to do with integrity check and have not taken into consideration dynamic trust analysis and real-time response systems. When considering machine learning security, several research works have demonstrated that the machine learning systems are largely susceptible to data poisoning. Barreno et al., Biggio and Roli, and Goodfellow et al. studied the security vulnerability of machine learning models and highlighted the importance of corrupted training data in causing a significant decline in model performance [24]. Papernot et al. also showed that even black-box attacks can be used to attack ML systems.

These publications demonstrate that it is necessary to protect not only the data but also the process of training. Yet, the majority of these researches concentrate on attack strategies, as opposed to suggesting powerful mechanisms of data integrity assurance. Recent studies have tried to combine blockchain and machine learning processes. Chen et al. and Yuan et al. suggested blockchain-based trusted machine learning systems to improve the transparency and security [25].

Salah et al. and Alshamsi et al. have addressed the functionality of blockchain to establish trustful AI and audit trail [26]. Though these studies enhance traceability, they are typically not fine-grained in dataset version control, dynamically trusting and do not explicitly give a linkage between dataset integrity and training outputs, restricting their usefulness in practice. Analytically, based

on the taxonomy of Bloom, there are critical questions that can be suggested on existing literature: What are the shortcomings of the current blockchain-based integrity models? Why are not the existing systems able to quantify trust dynamically? What can be done to identify dataset integrity violations as soon as possible and respond to them? Examination of existing literature indicates that the majority of systems consider integrity to be a dichotomy (authentic or inauthentic) and that they do not consider attenuation and restoration of trust (gradually).

Also, warning systems and automated response plans are not part of any current models. The research problem identified is, therefore, the lack of a single, trust-sensitive, and response-based framework of managing dataset integrity. Although blockchain provides immutability and Merkle Trees are efficient to verify, the existing existing solutions lack such functionality as trust scoring, role-based governance, variable backup policy, and security of machine learning training results. Such a gap is quite critical in the data sensitive environments where prompt action can be disastrous based on the delayed identification or un-verified datasets. The given approach can be explained by the fact that it bridges the gap between overall and integrated design. With the addition of a trust score calculation on each source of the data, the system surpasses the data integrity checks and it can now make risk-based decisions. The purpose of smart alert mechanism is to provide a fast reaction to integrity breaches, whereas backup of data sets differentially will guarantee the new storage and faster recovery.

Role-based access control and digital signatures enhance non-repudiation and tying the results of training the MLs to the established verified versions of the datasets directly combats the problem of trust as noted in earlier studies on the concept of ML security. Regarding tools and techniques, this research utilizes cryptographic hash algorithms that are standardized by NIST [27], Merkle Tree-based integrity checking with Merkle trees [28], and blockchain networks, including Ethereum and smart contracts used to store data impartially and automatically [29]. Blockchain development environments like Hardhat [30] are used to support the development and experimentation. The methodology of the evaluation involves the tampering tests on controlled datasets, integrity verification tests, trust score analysis and validation of linkage between datasets and machine learning products.

3. METHODOLOGY

The step-by-step study conduction procedure begins with dataset source registration, where each source is assigned an initial trust score and cryptographic identity. Datasets are selected and preprocessed by further dividing them into blocks, and cryptographic hash generated for every block. These hashes of each block formed into a single

hash and that is the hash value of the merkle root. During verification, the system again recomputes the values of hash and compares the newly generated Merkle root with the blockchain-stored value. Based on the result, the trust score is automatically updated, and alert generation, access restrictions are triggered.

Initial prototype testing was conducted as a proof to validate the feasibility of the proposed design. Controlled datasets of different sizes were used to simulate realistic operational scenarios. In initial tests, datasets were uploaded and verified without any modification, which produces correct Merkle Tree construction and successful block chain verification. In further tests, specific data blocks were tampered to observe the system behavior. The integrity verification mechanism successfully detects tampering, triggered severity-based alerts, reduces trust scores, and initiated differential backups. Additional tests validated role-based access control by restricting unauthorized operations (by using role based access control) and confirmed dataset ownership using digital signature verification. These prototype experiments demonstrated that the design principles could be effectively translated into a working system.

The prototype includes modules for dataset extraction, hash computation, Merkle Tree construction, block chain interaction, trust score management, alert generation, and access control management. Experimental demonstrations involved dataset uploads, version updates, integrity verification requests, and tampering simulations. The system successfully demonstrated real-time detection of integrity violations, trust score adjustments, and automated alert notifications. The linkage between dataset versions and machine learning models was also implemented to verify training authenticity, demonstrating practical applicability in sensitive AI workflows.

3.1. System Model

Let a dataset be represented as:

$$D = \{f_1, f_2, f_3, \dots, f_n\}$$

where f_i represents individual files.

The system performs:

Hash generation, Merkle Tree construction

Blockchain anchoring, Verification before training

3.2. Cryptographic Hashing

Each file is processed using SHA-256:

$$h_i = \text{SHA256}(f_i)$$

Properties:

- Fixed length output (256 bits)
- Collision resistant
- Sensitive to small changes

Dataset hash set: $H = \{h_1, h_2, \dots, h_n\}$

If any file is modified:

$$\text{SHA 256}(f_i') \neq h_i$$

Tampering is detected immediately.

3.3. Merkle Tree Construction

Hashes are combined pairwise:

$$h_{i,j} = \text{SHA 256}(h_i || h_j)$$

This process continues until a single root is obtained:

$$\text{MR} = \text{MerkleRoot}(H)$$

Merkle Root represents the entire dataset integrity.

Verification complexity:

$$O(\log n)$$

This makes the system scalable for large datasets.

3.4. Block chain Storage Model

The Merkle Root is stored using a smart contract.

Let: $B = \{\text{MR}, t, u\}$

Where

MR = Merkle Root, t = timestamp, u = uploader

Once stored:

$$\text{MR}_{\text{stored}} = \text{immutable}$$

Any modification will produce:

$$\text{MR}_{\text{new}} \neq \text{MR}_{\text{stored}}$$

3.5. Dataset Verification Algorithm

Algorithm:

Input: Dataset D

Compute current hashes, Generate MR_{new} , Retrieve $\text{MR}_{\text{stored}}$, Compare from blockchain

Verify = {Valid($\text{MR}_{\text{new}} = \text{MR}_{\text{stored}}$) /
Tampered ($\text{MR}_{\text{new}} \neq \text{MR}_{\text{stored}}$)}

3.6. Trust Score Model

Each contributor is assigned a trust score.

Let: $TS = \frac{V}{V+T}$

where

- V = successful verifications
- T = tampering incidents

Score range: $0 \leq TS \leq 1$

Trust levels:

- High: $TS > 0.8$
- Medium: $0.5 < TS \leq 0.8$
- Low: $TS \leq 0.5$

4. SYSTEM ARCHITECTURE

The process starts with the dataset upload, where the user submits the dataset into the system. At this stage, the data is temporarily stored for integrity. In next step, a SHA-256 cryptographic hash is generated. This hash acts as a unique fingerprint for the dataset. After hashing, the dataset is split into smaller blocks. Dividing the data into these smaller blocks makes it easier to create hash values for each of these small blocks. These block-level hashes are then used to create a Merkle Tree. In this structure, pairs of hashes are combined and re-hashed repeatedly until a single hash, known as the Merkle Root. The generated Merkle Root is stored on the blockchain.

Since blockchain records are immutable and tamper-proof, storing the Merkle Root ensures dataset without exposing the actual data on-chain. Next, the dataset is reprocessed to regenerate the Merkle Root and this value is compared with the Merkle Root stored on the blockchain. If both values match, the dataset is confirmed to be authentic and unchanged. Finally, the process ends with a verified and trusted dataset.

Layers:

- Frontend(React)
- Backend(Node.js)
- Verification Engine
- Blockchain Layer
- AI Training Module



Fig-1: Flow chart

5. PERFORMANCE ANALYSIS

Table -3.1: Trust Score Evolution

Test ID	Condition	Trust Score Action
TS01	New source registration	Score initialized to 50
TS02	Successful verification	Score increased
TS03	Tampering detected	Score decreased
TS04	Dashboard view	Score displayed

Each data source is assigned an initial trust score at the time of registration. The score is updated dynamically based on dataset verification results. When a dataset passes integrity verification, the trust score is increased. If tampering or integrity failure is detected, the score is decreased.

The trust score update is defined as:

$$TS_{new} = TS_{old} + \alpha(\text{success})$$

$$TS_{new} = TS_{old} - \beta(\text{tampering})$$

Based on the score, sources are categorized as:

- High Trust : $TS > 80$
- Medium Trust : $50 \leq TS \leq 80$
- Low Trust : $TS < 50$

The updated score is displayed on the dashboard for monitoring source reliability.

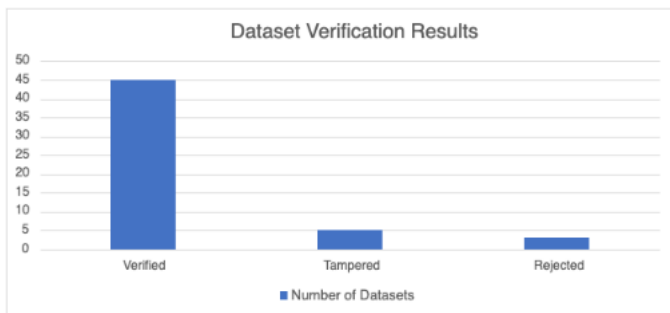


Fig-2: Dataset Verification Results

Fig. 2 shows the dataset verification results. Out of 53 datasets, 45 were verified successfully, while 5 tampered and 3 invalid datasets were detected and rejected, demonstrating effective integrity validation.

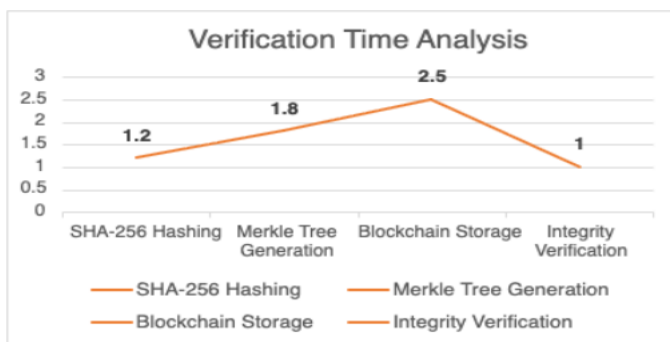


Fig -3: Verification Time Analysis

The time taken for stages of verifying the dataset is shown in Fig. 3. We broke down the processing time into four main operations: SHA-256 hashing, generating the Merkle Tree storing on the blockchain and checking data integrity. Out of these storing on the blockchain took the time at 2.5 seconds because it involves executing transactions and interacting with smart contracts. Generating the Merkle Tree took 1.8 seconds. SHA-256 hashing took 1.2 seconds. Checking data integrity took 1.0 second.

These results show that our system verifies data within a time making it practical, for real-world use. Most of the effort goes into blockchain operations but the cryptographic and verification processes are still efficient.

6. CONCLUSION

The main expectation of the study was to create and verify a safe, transparent and scalable framework that can guarantee integrity of data and trust in distributed systems. The experiment hypothesized that blockchain with Merkle Tree-based verification would provide an

effective way to detect data tempering and an efficient way to check large data sets. The experimental findings showed that the suggested system was capable of detecting even subtle changes in the dataset by the means of Merkle root mismatch, whereas blockchain storage made the system immutable and auditable. The trust score system worked as desired in that the reliability rating of the information sources in the dataset was dynamically changed according to the verification results. Moreover, the smart alert system was able to decrease the response time between tampering detection and administrative action and the differential dataset backups reduced significantly the storage overhead as well as the recovery time. The importance of this is due to the decentralized and tamper-resistant nature of blockchain, which removes the single-point failures, and the efficacy of Merkle trees which enable scalable verification without a lot of unnecessary computing.

7. FUTURE WORK

To make the proposed system even better we can consider some improvements. Here are some ideas:

- We can connect the system to public block chain networks to make the system more decentralized.
- We can use cloud-based storage to handle datasets in the system.
- We can create automated machine learning pipelines that verify the data in the system automatically.
- We can make the system work with time and streaming data for the Verichain framework.
- We can use machine learning models to evaluate trust in the Verichain framework.
- We can add techniques that preserve privacy like learning or zero-knowledge proofs, to the Verichain framework.

These improvements will make the Verichain framework more efficient and more scalable and more practical to use for people who use the Verichain framework.

8. REFERENCES

- [1] Jian-Xin, X. U., and Hou Zhong-Sheng. "Notes on data-driven system approaches." *Acta Automatica Sinica* 35.6 (2009): 668-675.
- [2] Craigen, Dan, Nadia Diakun-Thibault, and Randy Purse. "Definingcybersecurity." *Technologyinnovation management review* 4.10 (2014).
- [3] Figueiredo, Ana Sofia. "Data sharing: convert challenges into opportunities." *Frontiers in public health* 5 (2017): 327.
- [4] Stylianidis, Efstratios. "Trustworthy Data." *Exploring the Ethical Dimension in Recording and Documenting*

Cultural Heritage. Cham: Springer Nature Switzerland, 2025. 105-119.

[5] Grubb, Michael D., et al. "Sending out an sms: Automatic enrollment experiments for overdraft alerts." *The Journal of Finance* 80.1 (2025): 467-514.

[6] Mangel, Simon, et al. "Data reliability and trustworthiness through digital transmission contracts." *European Semantic Web Conference*. Cham: Springer International Publishing, 2021.

[7] Meyer, Georg, et al. "A machine learning approach to improving dynamic decision making." *Information systems research* 25.2 (2014): 239-263.

[8] Attaallah, Abdulaziz, Abdullah Algarni, and Raees Ahmad Khan. "Managing Security-Risks for Improving Security-Durability of Institutional Web-Applications: Design Perspective." *Computers, Materials & Continua* 66.2 (2021).

[9] Sobti, Rajeev, and Ganesan Geetha. "Cryptographic hash functions: a review." *International Journal of Computer Science Issues (IJCSI)* 9.2 (2012): 461.

[10] Liu, Haojun, et al. "Merkle tree: A fundamental component of blockchains." 2021 *International Conference on Electronic Information Engineering and Computer Science (EIECS)*. IEEE, 2021.

[11] Chanchev, Ivaylo. "Speedup of Merkle-Root Hash Value Computation Using Groups." *Future of Information and Communication Conference*. Cham: Springer Nature Switzerland, 2023.

[12] Albarrak, Reem M., and Daniel A. Menasce. "Trust but verify: A framework for the trustworthiness of distributed systems." *IEEE Transactions on Dependable and Secure Computing* 19.3 (2020): 2105-2121.

[13] Kuznetsov, O., Rusnak, A., Yezhov, A., Kuznetsova, K., Kanonik, D., & Domin, O. (2024). Merkle trees in blockchain: A study of collision probability and security implications. *Internet of Things*, 26, 101193.

[14] Jafargholi, Zahra, and Daniel Wachs. "Tamper detection and continuous non-malleable codes." *Theory of Cryptography Conference*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015.

[15] Ferraiolo, David, D. Richard Kuhn, and Ramaswamy Chandramouli. *Role-based access control*. Artech house, 2003.

[16] Mohan, Arun Prasad, and Angelin Gladston. "Merkle tree and blockchain-based cloud data auditing." *International Journal of Cloud Applications and Computing (IJCAC)* 10.3 (2020): 54-66.

[17] Wei, PengCheng, et al. "Blockchain data-based cloud data integrity protection mechanism." *Future Generation Computer Systems* 102 (2020): 902-911.

[18] Chelladurai, Usharani, and Seethalakshmi Pandian. "Hare: A new hash-based authenticated reliable and efficient modified merkle tree data structure to ensure integrity of data in the healthcare systems." *Journal of Ambient Intelligence and Humanized Computing* (2021): 1-15.

[19] Senarathna, Janaka Ishan. "The Role of Cryptography in Blockchain: Ensuring Immutability, Transparency and Security." (2025).

[20] Krichen, Moez. "Improving formal verification and testing techniques for internet of things and smart cities." *Mobile networks and applications* 28.2 (2023): 732-743.

[21] Platt, Moritz, et al. "Information privacy in decentralized applications." *Trust models for next-generation blockchain ecosystems*. Cham: Springer International Publishing, 2021. 85-104.

[22] H. Hasan and K. Salah, "Blockchain-based provenance model for data integrity," *IEEE Access*, vol. 6, pp. 21271-21283, 2018.

[23] He, Kai, et al. "Blockchain based data integrity verification for cloud storage with T-merkle tree." *International Conference on Algorithms and Architectures for Parallel Processing*. Cham: Springer International Publishing, 2020.

[24] Sharma, Brihat, Chandra N. Sekharan, and Fanyu Zuo. "Merkle-tree based approach for ensuring integrity of electronic medical records." 2018 9th *IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*. IEEE, 2018.

[25] Ural, Ozgur, and Kenji Yoshigoe. "Survey on blockchain-enhanced machine learning." *IEEE Access* 11 (2023): 145331-145362.

[26] AlShamsi, Mohammed, et al. "Artificial intelligence and blockchain for transparency in governance." *Artificial intelligence for sustainable development: Theory, practice and future applications*. Cham: Springer International Publishing, 2020. 219-230.

[27] Brandão, Luís TAN, et al. "NIST roadmap toward criteria for threshold schemes for cryptographic primitives." 7 Jul. 2020,

[28] Sharma, Brihat, Chandra N. Sekharan, and Fanyu Zuo. "Merkle-tree based approach for ensuring integrity of electronic medical records." 2018 9th *IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*. IEEE, 2018.

[29] Batchu, Sai, et al. "Using ethereum smart contracts to store and share COVID-19 patient data." *Cureus* 14.1 (2022).

[30] Naik, Poornima G., and Girish R. Naik. *Every Stuff You Need for Development of Decentralized App Using Blockchain Technology:(Covers Hardhat, React. js and Ethers. js)*. Shashwat Publication, 2023.