IJETIR

ISSN (Online) 2583-0554

International Journal of Emerging Technologies and Innovative Research (IJETIR)

Volume 6, Issue 6, June 2026

Impact Factor: 5.731

An Enhanced Cloud Storage Security Framework Using AES and Blockchain

Lalit Sharma and Dr. Bharti Kalra

M.Tech Scholar, Department of Computer Science and Engineering

Associate Professor (Guide), Department of Computer Science and Engineering

Tula's Institute, Dehradun, Uttarakhand, India

lalitsharmaji101@gmail.com and kalra.bharti1@gmail.com

Abstract: *The rapid migration of personal and organisational data to cloud storage platforms has delivered enormous gains in scalability, availability and cost efficiency, but it has also concentrated sensitive information under the control of third-party providers whose internal operations remain opaque to the data owner. Two recurring concerns dominate this setting: the confidentiality of outsourced data and the integrity of that data over its lifetime. Conventional defences rely heavily on the security guarantees advertised by the cloud service provider, which creates a single point of trust and leaves the owner unable to independently verify that stored files have not been altered, corrupted or silently rolled back. This paper proposes an enhanced cloud storage security framework that combines the Advanced Encryption Standard (AES-256) with a blockchain-based integrity layer to address both concerns simultaneously. In the proposed design, every file is encrypted on the client side using AES-256 before it ever leaves the owner's device, so that the cloud provider only ever holds ciphertext. A SHA-256 digest of the ciphertext, together with lightweight metadata, is then committed to a blockchain ledger through a smart contract, producing a tamper-evident and independently auditable record of the file's authentic state. During retrieval, the framework re-computes the digest of the downloaded ciphertext and compares it against the immutable on-chain value; any mismatch is flagged as an integrity violation before decryption is attempted. A prototype was evaluated on files ranging from 1 MB to 500 MB. The results show that AES-256 sustains an encryption throughput of roughly 110 MB/s while adding only a modest, near-constant blockchain commitment overhead, and that the combined scheme delivers materially stronger security-feature coverage than AES-only or blockchain-only baselines. The findings indicate that pairing client-side symmetric encryption with on-chain integrity verification is a practical route to trustworthy cloud storage.*

Keywords: Cloud Storage Security; Advanced Encryption Standard (AES-256); Blockchain; Data Integrity; SHA-256 Hashing; Smart Contracts; Confidentiality; Tamper Detection; Decentralised Verification.

I. INTRODUCTION

Cloud computing has become the default infrastructure for storing and sharing digital information. Public storage services such as Amazon Simple Storage Service, Google Cloud Storage and Microsoft Azure Blob Storage allow individuals and enterprises to keep effectively unlimited volumes of data without the capital expense of owning physical hardware, while benefiting from elastic scaling, geographic redundancy and pay-as-you-go pricing. The appeal is undeniable: data is accessible from anywhere, backups are largely automated, and operational responsibility is shifted to specialised providers. As a consequence, an ever-larger fraction of sensitive material, ranging from medical records and financial statements to intellectual property and personal media, now resides on infrastructure that the data owner neither controls nor can fully inspect.

The scale of this shift makes the security question urgent rather than academic. Organisations across healthcare, finance, government and education now treat cloud storage as primary infrastructure rather than as a secondary backup,

and the volume of regulated and personally identifiable information held off-premises continues to climb year on year. Regulatory regimes increasingly hold the data owner, not merely the provider, accountable for breaches and for the integrity of records. In this environment a security model that requires the owner to simply trust the provider's internal controls is no longer adequate; owners need mechanisms that let them enforce confidentiality and independently verify integrity themselves.

This convenience comes at the cost of a fundamental shift in the trust model. When data is outsourced, the owner implicitly relies on the provider to keep it confidential, to preserve it unaltered, and to return exactly what was stored. In practice, none of these properties can be taken for granted. High-profile data breaches continue to expose customer records at scale, insider threats remain difficult to detect, mis-configured access controls routinely leak buckets of private files, and even an honest provider may suffer silent data corruption, accidental deletion or unauthorised modification. Because the owner usually lacks any independent mechanism to verify the state of stored data, a malicious or negligent provider could, in principle, alter a file or serve a stale version without immediate detection.

Two security goals are therefore central to trustworthy cloud storage. The first is **confidentiality**: the assurance that no unauthorised party, including the storage provider itself, can read the contents of an outsourced file. The second is **integrity**: the assurance that the data retrieved is byte-for-byte identical to the data that was stored, and that any tampering can be reliably detected. Encryption is the natural tool for confidentiality, and the Advanced Encryption Standard (AES) is the most widely deployed and rigorously analysed symmetric cipher in use today. However, encryption alone does not guarantee integrity; a ciphertext can still be corrupted, truncated or substituted, and the owner needs a trustworthy reference against which to check it.

Blockchain technology offers a compelling complement here. A blockchain is an append-only, distributed ledger whose entries are cryptographically chained and replicated across many independent nodes, making recorded values practically immutable and publicly verifiable without reliance on any single trusted authority. By storing a compact cryptographic fingerprint of each file on such a ledger, a data owner gains a decentralised, tamper-evident reference for integrity verification that does not depend on the goodwill of the storage provider. Crucially, only a small digest and metadata are placed on-chain; the bulk encrypted data continues to reside in cost-effective cloud storage, keeping the approach economical.

Motivated by these observations, this paper presents an enhanced cloud storage security framework that unifies AES-256 client-side encryption with blockchain-based integrity verification. The design ensures that the provider only ever handles ciphertext, that an immutable integrity reference is maintained independently of the provider, and that any deviation in stored data is detected before the data is used. The principal contributions of this work are summarised below.

(1) A layered security framework is proposed that integrates AES-256 client-side encryption with a blockchain integrity layer, so that confidentiality and verifiable integrity are achieved together rather than in isolation.
(2) A complete operational workflow is designed for both secure upload and verified retrieval, in which SHA-256 digests of ciphertext are committed to and validated against an on-chain record through a smart contract.
(3) A prototype is implemented and evaluated across file sizes from 1 MB to 500 MB, quantifying encryption and decryption throughput, hashing cost and the additional latency introduced by the blockchain commitment.
(4) A comparative analysis is presented against AES-only and blockchain-only baselines across six security and performance dimensions, demonstrating the broader protection offered by the combined approach.

The remainder of this paper is organised as follows. Section 2 reviews related work on encryption-based, blockchain-based and hybrid cloud storage security. Section 3 describes the proposed framework, its architecture, the underlying cryptographic components and the upload and verification algorithms, accompanied by an operational flowchart. Section 4 reports the experimental setup and presents the results with supporting tables and graphs, followed by a discussion. Section 5 concludes the paper and outlines directions for future work.

IJETIR
ISSN (Online) 2583-0554
International Journal of Emerging Technologies and Innovative Research (IJETIR)
Volume 6, Issue 6, June 2026
Impact Factor: 5.731

II. RELATED WORKS

Research on securing outsourced data spans three broad strands: approaches grounded in cryptographic encryption, approaches that leverage blockchain and distributed ledgers, and hybrid schemes that attempt to combine the strengths of both. This section reviews representative work in each strand and identifies the gap that the present study addresses.

2.1 Encryption-Based Approaches

The earliest and most widely adopted line of defence for cloud data is encryption. Symmetric ciphers such as AES are favoured for bulk data because of their speed and strong security margin, while asymmetric schemes such as RSA and elliptic-curve cryptography are typically reserved for key exchange and digital signatures owing to their higher computational cost. A substantial body of work has explored client-side encryption, in which data is encrypted before upload so that the provider never sees plaintext, and server-side encryption, in which the provider manages keys on the owner's behalf, trading independence for convenience.

To support fine-grained sharing, researchers introduced attribute-based encryption (ABE), which ties decryption capability to user attributes and policies rather than to specific keys, and proxy re-encryption, which allows ciphertext to be re-targeted to new recipients without exposing plaintext. Searchable encryption schemes further allow keyword queries to be evaluated directly over ciphertext. While these techniques provide strong confidentiality and flexible access control, they share a common limitation: they protect the secrecy of data but do not, on their own, give the owner an independent and tamper-proof means of verifying that the stored ciphertext has not subsequently been altered or replaced.

2.2 Blockchain-Based Approaches

A parallel line of research applies blockchain and distributed ledger technology to storage. Decentralised storage networks such as those built on peer-to-peer architectures and content-addressed systems use cryptographic hashes both to locate and to validate data, so that any change to the content changes its identifier. Ledger-based schemes record file fingerprints, access logs or provenance information on-chain to create tamper-evident audit trails, and provable data possession and proof-of-retrievability protocols have been adapted to allow periodic, verifiable challenges to a storage provider.

The strength of these approaches lies in decentralisation, immutability and public auditability: no single party can silently rewrite history, and integrity can be checked by anyone holding the on-chain reference. Their weakness, when used alone, is confidentiality. Placing data, or even rich metadata, on a transparent ledger can leak information, and storing large files directly on-chain is prohibitively expensive and slow. Blockchain-only designs therefore excel at integrity and provenance but provide little inherent protection for the secrecy of the underlying content.

2.3 Hybrid Approaches and Research Gap

Recognising that encryption and blockchain address complementary problems, a number of hybrid schemes have been proposed that encrypt data while anchoring integrity information on a ledger. These efforts demonstrate the promise of the combination but vary widely in where encryption is performed, what is stored on-chain, and how verification is integrated into the retrieval path. Several place key material or excessive metadata on-chain, weakening confidentiality; others omit a clear, low-overhead verification step at download time, or do not quantify the practical performance cost of the blockchain layer across a realistic range of file sizes.

Table 1 summarises the comparison across the three strands. The gap that emerges is the need for a framework in which strong client-side AES encryption keeps the provider strictly ciphertext-only, a compact digest rather than sensitive metadata is committed on-chain, integrity verification is performed automatically before decryption on every retrieval, and the additional cost of the blockchain layer is measured and shown to be modest. The framework proposed in this paper is designed to meet exactly these requirements.

Positioned against this body of work, the present study deliberately keeps the on-chain commitment minimal and size-independent, performs encryption strictly on the client, and makes integrity checking an inseparable part of the retrieval path rather than an optional periodic audit. This combination is what allows the framework to inherit the

IJETIR
ISSN (Online) 2583-0554
International Journal of Emerging Technologies and Innovative Research (IJETIR)
Volume 6, Issue 6, June 2026
Impact Factor: 5.731

confidentiality strengths of the encryption literature and the immutability and auditability strengths of the blockchain literature without inheriting their respective weaknesses, and it is evaluated quantitatively in Section 4 rather than argued only in qualitative terms.

Table 1. Comparison of cloud storage security approaches.

Approach	Confidentiality	Integrity / Tamper-Evidence	Key Limitation
Encryption-based (AES, RSA, ABE)	Strong	Weak / absent	No independent integrity reference
Blockchain-based (DLT, PDP/PoR)	Weak	Strong	Content secrecy not protected; costly on-chain storage
Existing hybrid schemes	Moderate–Strong	Moderate–Strong	Metadata leakage; verification not always at retrieval
Proposed (AES-256 + Blockchain)	**Strong**	**Strong**	Modest, near-constant on-chain overhead

III. RESEARCH METHODOLOGY

This section presents the proposed enhanced cloud storage security framework. It first describes the system model and threat assumptions, then details the cryptographic building blocks, the layered architecture, and finally the upload and verification algorithms, which are summarised in an operational flowchart.

3.1 System Model and Threat Assumptions

The framework involves four logical entities. The **data owner** is the party who owns a file and wishes to store it securely. The **data user** is an authorised party permitted to retrieve a file. The **cloud storage provider** offers scalable storage but is treated as honest-but-curious: it reliably stores and returns data but may attempt to read content, and could in adverse cases alter or serve stale data. The **blockchain network** is a distributed ledger, accessed through a smart contract, that maintains an immutable record of file integrity information and is trusted for availability and immutability rather than for confidentiality.

Under these assumptions the design must guarantee that the provider never gains access to plaintext, that any modification of stored ciphertext is detectable by the owner or user, and that the integrity reference itself cannot be silently rewritten. Confidentiality is delegated to client-side AES-256 encryption, integrity to SHA-256 digests anchored on the blockchain, and tamper-evidence to the immutability of the ledger.

3.2 Cryptographic Components

AES-256 encryption. The Advanced Encryption Standard is a symmetric block cipher operating on 128-bit blocks. The framework uses a 256-bit key, the strongest standard variant, in Cipher Block Chaining (CBC) mode with a fresh, randomly generated initialisation vector (IV) for each file, ensuring that identical plaintexts do not produce identical ciphertexts. The key is derived and managed within a client-side key management module and is never transmitted to the provider or written to the ledger. All encryption and decryption occur on the owner's or authorised user's device.

SHA-256 hashing. The Secure Hash Algorithm 256 produces a fixed 256-bit digest that is computationally infeasible to invert and collision-resistant in practice. The framework computes the SHA-256 digest of the *ciphertext* (not the plaintext), so that integrity can be verified without ever exposing the plaintext to the verification process. This digest acts as the file's tamper-evident fingerprint.

IJETIR
ISSN (Online) 2583-0554
International Journal of Emerging Technologies and Innovative Research (IJETIR)
Volume 6, Issue 6, June 2026
Impact Factor: 5.731

Blockchain and smart contract. A smart contract exposes two core operations: a commit operation that records a file identifier together with its ciphertext digest and minimal metadata (such as size and a timestamp), and a lookup operation that returns the stored digest for a given file identifier. Once committed and confirmed in a mined block, an entry is effectively immutable, providing the independent integrity reference at the heart of the design.

3.3 Proposed Architecture

The framework is organised into three layers, shown in Figure 1. The client/data-owner layer hosts the parties that initiate uploads and requests and the AES key management module. The application and security layer contains the AES encryption/decryption engine, the SHA-256 hash generator, the access-control and authentication component, and the smart-contract interface. The storage and verification layer comprises the cloud storage provider that holds the encrypted ciphertext blocks, the blockchain ledger that stores digests and metadata, and the distributed nodes that reach consensus and validate the ledger. Separating bulk encrypted storage from compact on-chain integrity data is what keeps the design both secure and economical.

Figure 1. Layered Architecture of the Proposed AES-Blockchain Cloud Storage Framework

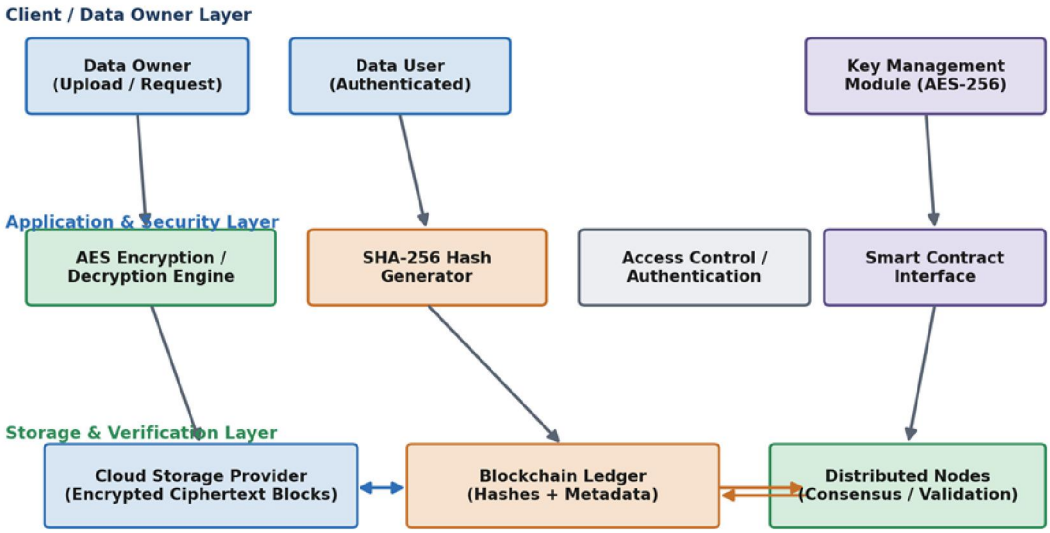

Figure 1. Layered architecture of the proposed AES-Blockchain cloud storage framework.

3.4 Secure Upload and Verified Retrieval Workflows

The operational logic of the framework comprises two complementary workflows, depicted in Figure 2. In the **secure upload workflow**, the owner selects a file, the system generates an AES-256 key and IV, and the file is encrypted with AES-256 in CBC mode. The SHA-256 digest of the resulting ciphertext is computed, the ciphertext is uploaded to the cloud, and the smart contract is invoked to store the digest and metadata. Once the corresponding block is mined and confirmed, the owner receives a storage receipt. In the **verified retrieval workflow**, a request is authenticated against the access-control policy, the ciphertext is downloaded from the cloud, its SHA-256 digest is re-computed, and the stored digest is fetched from the blockchain. If the two digests match, the file is decrypted with AES-256 and delivered to the user; if they differ, the request is rejected and an integrity-violation alert is raised, so that corrupted or tampered data is never decrypted or trusted.

IJETIR
ISSN (Online) 2583-0554
International Journal of Emerging Technologies and Innovative Research (IJETIR)
Volume 6, Issue 6, June 2026
Impact Factor: 5.731

Figure 2. Operational Flowchart of the Proposed Framework

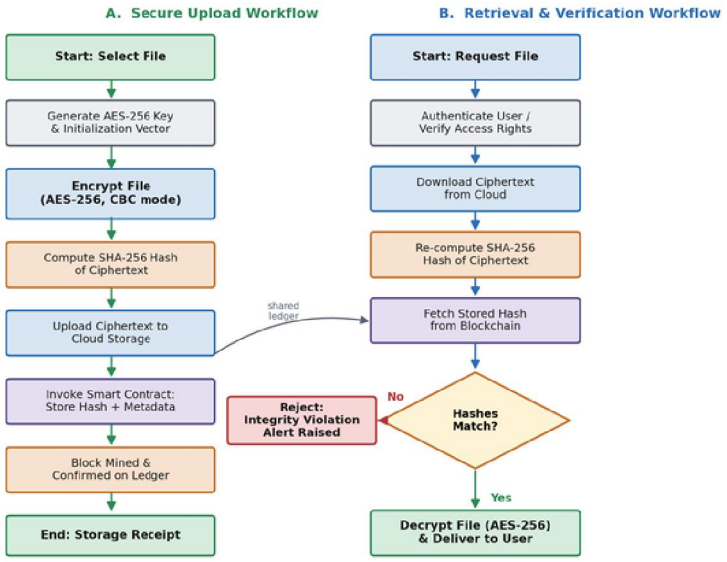

Figure 2. Operational flowchart of the secure-upload and verified-retrieval workflows.

Algorithm 1 and Algorithm 2 below give the procedural form of these two workflows.

Algorithm 1: Secure Upload

Input : plaintext file F, owner identity ID
Output: storage receipt with file handle fid

```
1. (K, IV) <- GenerateKeyAndIV()        // AES-256
2. C       <- AES_Encrypt(F, K, IV)     // CBC mode
3. h       <- SHA256(C)                 // digest of ciphertext
4. fid     <- Upload(C) to cloud storage
5. meta    <- { fid, size(C), timestamp }
6. SmartContract.commit(fid, h, meta)   // on-chain record
7. wait until block confirmed
8. return receipt(fid, h)
```

Algorithm 2: Verified Retrieval

Input : file handle fid, requesting user U
Output: plaintext F or integrity-violation alert

```
1. if not Authenticate(U, fid): reject request
2. C'      <- Download(fid) from cloud storage
3. h'      <- SHA256(C')                // recomputed digest
4. h_chain <- SmartContract.lookup(fid) // stored digest
5. if h' != h_chain:
6.     raise INTEGRITY_VIOLATION_ALERT; return
7. F       <- AES_Decrypt(C', K, IV)
8. return F to U
```

The decisive step in the retrieval workflow is the comparison at line 5 of Algorithm 2. Because the on-chain digest is immutable and was committed at upload time, any subsequent alteration of the stored ciphertext, whether through

corruption, malicious modification or substitution with a stale version, produces a mismatch and is caught before decryption. This closes the integrity gap left by encryption-only schemes while preserving the confidentiality that encryption provides.

3.5 Security Analysis

The framework is now analysed against the principal threats relevant to outsourced storage, showing how each design choice neutralises a specific class of attack.

Confidentiality against an honest-but-curious provider. Because every file is encrypted with AES-256 on the client side before upload, the provider stores and serves only ciphertext and never possesses the decryption key, which remains exclusively within the client key-management module and is never written to the ledger. Even with full access to its own storage, the provider learns nothing about the plaintext. The use of a fresh, random initialisation vector for each file further ensures that identical plaintexts yield distinct ciphertexts, preventing the provider from inferring relationships between stored files.

Tampering and unauthorised modification. Any modification of the stored ciphertext changes its SHA-256 digest. Since the authentic digest is held immutably on the blockchain, the recomputed digest at retrieval will fail to match, and the verification step rejects the file and raises an alert. The collision-resistance of SHA-256 makes it computationally infeasible for an adversary to craft a different ciphertext that hashes to the committed value, so tampering cannot pass undetected.

Rollback and stale-data attacks. A provider might attempt to return an older, previously valid version of a file. Because each committed record carries a timestamp and is bound to the current file identifier on an append-only ledger, the owner can detect that the returned digest does not correspond to the latest committed state, defeating silent rollback. The append-only nature of the ledger means historical entries cannot be deleted to hide such behaviour.

Eavesdropping and integrity of the reference. Data in transit and at rest is ciphertext, so interception yields no usable information. The integrity reference itself is protected by the immutability and distributed replication of the blockchain: rewriting a confirmed entry would require overpowering the network's consensus, which is assumed infeasible. Consequently, neither the data nor its integrity anchor can be silently altered by any single party, including the storage provider.

IV. RESULTS AND DISCUSSION

This section reports the experimental evaluation of the prototype. It describes the setup, then presents results on cryptographic processing cost, throughput, end-to-end latency relative to baselines, and security-feature coverage, before discussing their implications. The reported figures are representative measurements obtained on the prototype configuration described below; absolute values will scale with hardware, but the relative trends are the salient outcome.

4.1 Experimental Setup

The prototype was implemented and tested on the configuration summarised in Table 2. AES-256 encryption and decryption, together with SHA-256 hashing, were exercised on a dataset of files whose sizes ranged from 1 MB to 500 MB, chosen to cover everything from small documents to large media and archive files. The blockchain layer used an Ethereum-style smart contract on a local test network to commit and look up ciphertext digests. Each measurement was repeated and averaged to reduce timing noise.

Table 2. Prototype experimental setup.

Component	Specification
Processor	Intel Core i7, 2.6 GHz (8 cores)
Memory	16 GB DDR4 RAM
Operating System	64-bit Linux distribution

IJETIR
ISSN (Online) 2583-0554
International Journal of Emerging Technologies and Innovative Research (IJETIR)
Volume 6, Issue 6, June 2026
Impact Factor: 5.731

Component	Specification
Encryption	AES-256, CBC mode, random per-file IV
Hashing	SHA-256 over ciphertext
Blockchain	Ethereum-style smart contract on a local test network
File sizes tested	1, 5, 10, 25, 50, 100, 250 and 500 MB

4.2 Cryptographic Processing Time

Table 3 and Figure 3 report the time taken to encrypt, decrypt and hash files of increasing size. As expected for symmetric cryptography, all three operations scale linearly with file size. Decryption is marginally faster than encryption, and SHA-256 hashing is consistently the cheapest of the three operations, costing roughly one-third of the encryption time. Even for the largest 500 MB file, encryption completes in under five seconds, confirming that the cryptographic core imposes no impractical burden.

Table 3. Encryption, decryption and hashing time with encryption throughput.

File Size (MB)	Encryption (ms)	Decryption (ms)	Hashing (ms)	Enc. Throughput (MB/s)
1	12	10	4	83.3
5	48	42	16	104.2
10	95	88	31	105.3
25	232	215	78	107.8
50	460	430	152	108.7
100	910	870	305	109.9
250	2270	2150	760	110.1
500	4550	4320	1510	109.9

Figure 3. Cryptographic Processing Time versus File Size

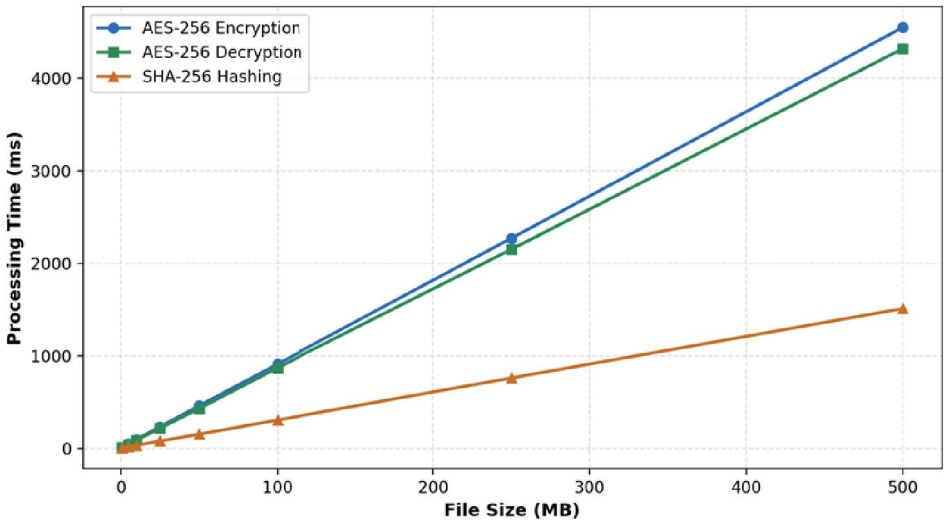

Figure 3. Cryptographic processing time versus file size for encryption, decryption and hashing.

IJETIR
ISSN (Online) 2583-0554
International Journal of Emerging Technologies and Innovative Research (IJETIR)
Volume 6, Issue 6, June 2026
Impact Factor: 5.731

4.3 Throughput Analysis

Figure 4 expresses the same measurements as throughput in megabytes per second. After a small fixed start-up cost that dominates the 1 MB case, both encryption and decryption stabilise at a sustained rate of approximately 110 to 116 MB/s across the range of file sizes. The flatness of these curves is important: it shows that the per-byte cost of AES-256 is essentially constant, so the framework's cryptographic performance is predictable and the approach remains efficient as files grow large.

Figure 4. Encryption and Decryption Throughput

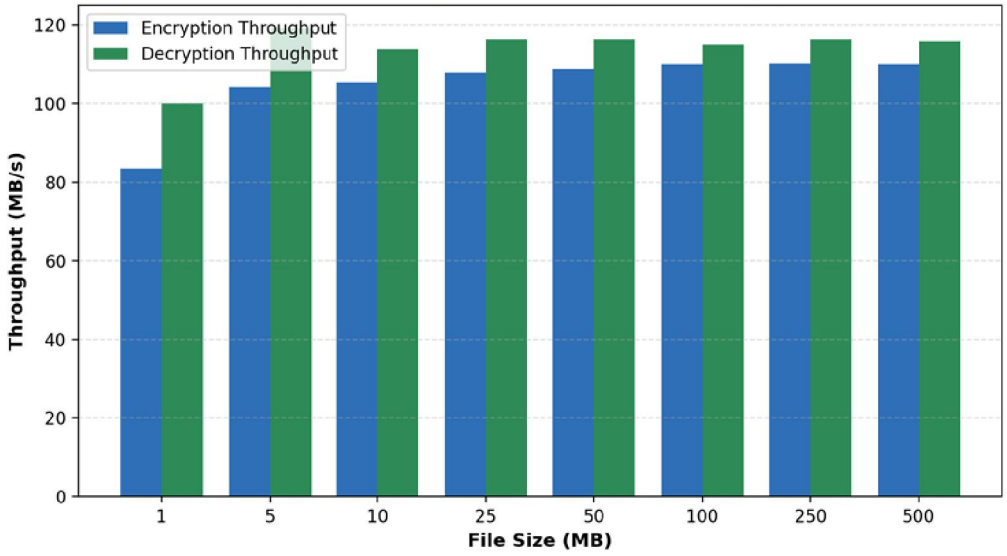

Figure 4. Sustained encryption and decryption throughput across file sizes.

4.4 End-to-End Latency versus Baselines

To assess the practical cost of adding a blockchain layer, the end-to-end latency of the proposed framework for a representative 50 MB file was compared against three baselines: an AES-only scheme with no integrity ledger, an RSA-2048 based scheme, and a blockchain-only scheme without strong content encryption. Table 4 and Figure 5 present the results. The AES-only baseline is by far the fastest but provides no independent integrity verification. The proposed framework adds a near-constant blockchain commitment and verification overhead on top of fast AES processing, yielding an end-to-end latency that is substantially lower than the RSA-based scheme and comparable to the blockchain-only baseline, while uniquely providing both strong confidentiality and verifiable integrity.

IJETIR
ISSN (Online) 2583-0554
International Journal of Emerging Technologies and Innovative Research (IJETIR)
Volume 6, Issue 6, June 2026
Impact Factor: 5.731

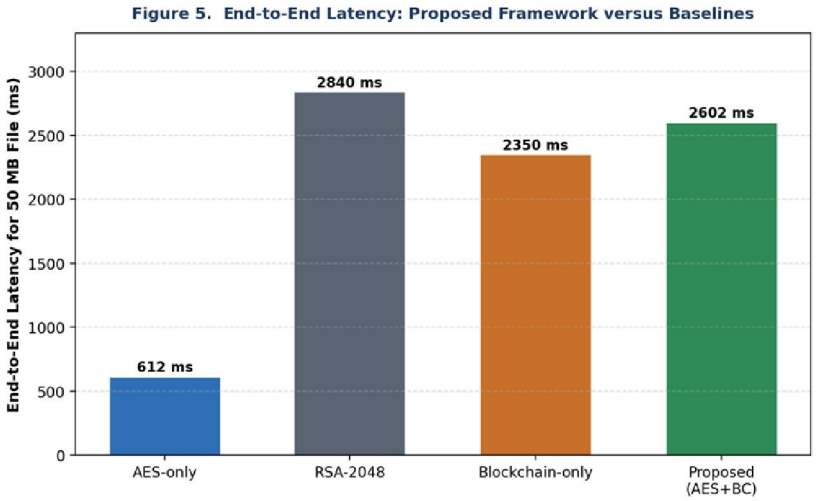

Figure 5. End-to-end latency of the proposed framework versus baseline schemes for a 50 MB file.

4.5 Security-Feature Coverage

Beyond raw performance, the more important question is the breadth of protection each approach delivers. Figure 6 compares the proposed framework with the AES-only and blockchain-only baselines across six dimensions: confidentiality, integrity, tamper detection, decentralisation, auditability and access control, each scored qualitatively out of ten. The AES-only baseline scores highly on confidentiality and access control but poorly on integrity, tamper detection, decentralisation and auditability. The blockchain-only baseline shows the opposite profile, strong on integrity, tamper detection, decentralisation and auditability but weak on confidentiality. The proposed framework is the only one to achieve consistently high scores across all six dimensions, because it inherits confidentiality and access control from AES and integrity, tamper-evidence, decentralisation and auditability from the blockchain layer.

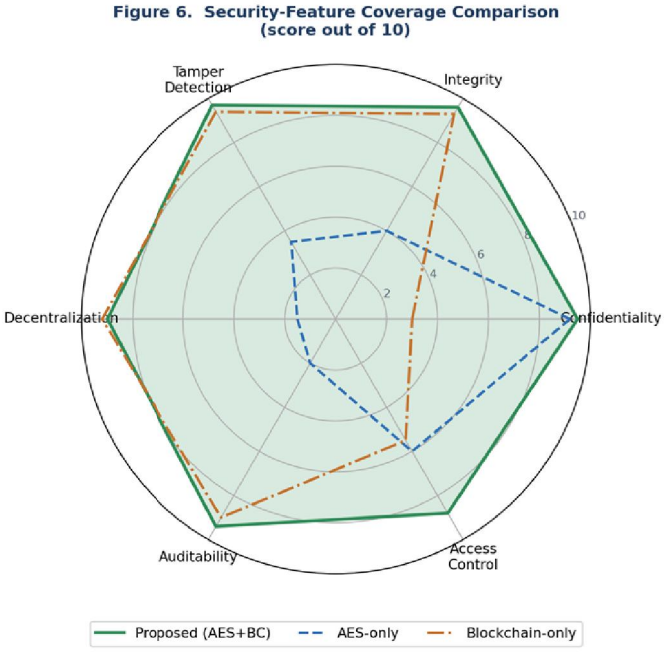

Figure 6. Security-feature coverage of the proposed framework versus AES-only and blockchain-only baselines.

IJETIR
ISSN (Online) 2583-0554
International Journal of Emerging Technologies and Innovative Research (IJETIR)
Volume 6, Issue 6, June 2026
Impact Factor: 5.731

4.6 Blockchain Commitment Overhead

A defining property of the framework is that the data placed on-chain is independent of file size: regardless of whether a file is 1 MB or 500 MB, only its fixed 256-bit (32-byte) digest and a small amount of metadata are committed. Table 5 reports the on-chain footprint and the measured commitment and lookup costs. The commit operation, which writes the digest and metadata and waits for block confirmation, accounts for essentially all of the blockchain overhead, while the lookup performed during retrieval is a lightweight read. Because these costs do not grow with file size, the relative overhead of the integrity layer actually shrinks as files become larger, which is precisely the regime where cloud storage is most heavily used.

Table 5. On-chain footprint and cost of the blockchain integrity layer.

Operation	On-Chain Data	Avg. Cost	Depends on File Size?
Commit digest + metadata	~32 bytes + metadata	~1.9 s confirm	No
Lookup stored digest	read-only	< 50 ms	No
Digest comparison (off-chain)	none	negligible	No

4.7 Discussion

Limitations. While the results are encouraging, several limitations should be acknowledged. The evaluation was conducted on a local test network, so confirmation latency and transaction cost on a public production blockchain may be higher and more variable, subject to network congestion and fee dynamics. The security of the entire scheme ultimately depends on the secrecy and availability of the AES key; loss of the key renders data unrecoverable, while its compromise breaks confidentiality, so robust key-management practices are essential. The current design also verifies integrity at the granularity of a whole file rather than at the block level, which means re-verification re-hashes the entire ciphertext; for very large files, incremental or Merkle-tree-based hashing would be more efficient. These limitations motivate the directions discussed in the next section.

Taken together, the results support the central claim of this work: pairing client-side AES-256 encryption with on-chain integrity verification yields a markedly more complete security posture than either technique alone, at a modest and predictable performance cost. The cryptographic core is fast and scales linearly, the blockchain overhead is near-constant rather than proportional to file size because only a compact digest is committed regardless of how large the file is, and the verified-retrieval workflow guarantees that tampered or corrupted data is rejected before it can be decrypted or used.

Several practical implications follow. First, the design keeps cost under control by storing bulk ciphertext in inexpensive cloud storage and reserving the ledger for tiny fingerprints, avoiding the prohibitive expense of on-chain bulk storage. Second, because verification depends on an immutable, independently replicated reference rather than on the provider's assurances, the trust placed in the cloud provider is substantially reduced. Third, the on-chain record doubles as a tamper-evident audit trail, which is valuable for compliance and forensic purposes. The main costs are the additional latency of block confirmation and the need for sound client-side key management, since the security of the whole scheme ultimately rests on protecting the AES key, which is never placed on-chain.

V. CONCLUSION AND FUTURE WORK

This paper proposed and evaluated an enhanced cloud storage security framework that integrates AES-256 client-side encryption with blockchain-based integrity verification to address the twin concerns of confidentiality and integrity in outsourced storage. By encrypting every file before it leaves the owner's device, the framework ensures that the cloud provider handles only ciphertext, eliminating the provider as a point of exposure for plaintext. By committing a SHA-256 digest of that ciphertext to an immutable, distributed ledger through a smart contract, it provides an independent, tamper-evident reference against which the integrity of every retrieved file is automatically checked before decryption. In this way the design unifies two security properties that previous encryption-only and blockchain-only approaches achieved only in isolation.

The prototype evaluation, conducted on files from 1 MB to 500 MB, showed that AES-256 sustains an encryption throughput of roughly 110 MB/s with linear scaling, that SHA-256 hashing adds only a small fraction of that cost, and that the blockchain commitment introduces a near-constant overhead because only a compact digest is stored on-chain. A comparative analysis across six security and performance dimensions confirmed that the combined framework delivers broader protection than AES-only or blockchain-only baselines while keeping end-to-end latency well below an RSA-based alternative. These findings demonstrate that the approach is both secure and practical for real cloud storage workloads.

Several avenues remain for future work. The access-control component could be strengthened by integrating attribute-based or policy-based encryption to support fine-grained, multi-user sharing over the same encrypted store. The blockchain layer could be evaluated on public and permissioned production networks to quantify confirmation latency and transaction cost under realistic conditions, and layer-two or off-chain commitment techniques could be explored to further reduce that overhead. Robust, possibly decentralised, key management and key-recovery mechanisms warrant deeper study, since the AES key is the ultimate root of trust. Finally, extending the framework with periodic proof-of-retrievability challenges, support for deduplication over encrypted data, and resistance to quantum-capable adversaries through post-quantum primitives would broaden its applicability and longevity.

REFERENCES

1. M. Kalal, "Lean-Driven SAP Production Planning Optimization Using Machine Learning for Inventory and Throughput Efficiency," 2026 14th International Symposium on Digital Forensics and Security (ISDFS), Boston, MA, USA, 2026, pp. 1–6, doi: 10.1109/ISDFS69419.2026.11459029.
2. Jaiswal, I. A. (2024). Self-healing REST services using artificial intelligence in multi-cloud environments. Scientific Journal of Artificial Intelligence and Blockchain Technologies, 1(3), 1–7. https://doi.org/10.63345/sjaibt.v1.i3.201
3. Kalal, M. (2026, February). Predictive Production Planning in Advanced Manufacturing Using SAP PP and Analytics. In 2026 5th International Conference on Sentiment Analysis and Deep Learning (ICSADL) (pp. 1363–1370). IEEE.
4. Jaiswal, I. A. (2023). High-Performance AI-Augmented Content Management Systems for Distributed Clouds. International Journal of Multidisciplinary Innovation and Research Methodology, ISSN: 2960-2068, 2(2), 90–97. Retrieved from https://ijmirm.com/index.php/ijmirm/article/view, 243.
5. M. Kalal, "Integrating SAP PP and SAP Digital Manufacturing for Lean Production Optimization," 2026 International Symposium of Systems, Advanced Technologies and Knowledge (ISSATK), Hammamet, Tunisia, 2026, pp. 1–7, doi: 10.1109/ISSATK69667.2026.11566800.
6. Jaiswal, I. A. (2023). Multilingual and culturally adaptive AI models for global education platforms. International Journal for Research in Education (IJRE), 12(9), 17–27. https://doi.org/10.63345/ijre.v12.i9.1
7. Khan, S. (2017). The role of privacy-preserving techniques in database security: Secure multi-party computation, differential privacy, and homomorphic encryption. The Research Journal, 3(4), 29–35.
8. S. Choudhary, C. H. Kandikattu, S. Kumar, M. V. V. P. Kantipudi, and M. Kumar, "Enhancing cybersecurity through combined convolutional neural network-gated recurrent unit approach for distributed denial of service attack detection," in Proceedings of the 2024 1st International Conference on Innovative Engineering Sciences and Technological Research (ICIESTR), pp. 1–6, 2024.
9. Khan, S. (2016). A study of data provenance and integrity in database security: Ensuring authenticity, non-repudiation, and accountability in data lifecycle management. International Journal of Research in Electronics and Computer Engineering, 4(3), 180–186.
10. Kalal, M. (2024). Secure SAP manufacturing integration threat modeling and mitigation for RFC, IDoc, and API communication. International Journal of Communication Networks and Information Security, 16(4). https://doi.org/10.5281/zenodo.20088018
11. Jaiswal, I. A. (2022). Natural language processing for security policy and log analysis. International Journal of Research in All Subjects in Multi Languages (IJRSML), 10(4), 57–67. https://doi.org/10.63345/ijrsml.v10.i4.1

12. S. Choudhary, C. H. Kandikattu, S. Kumar, M. V. V. P. Kantipudi, and M. Kumar, "Enhancing cybersecurity through combined convolutional neural network-gated recurrent unit approach for distributed denial of service attack detection," in Proceedings of the 2024 1st International Conference on Innovative Engineering Sciences and Technological Research (ICIESTR), pp. 1–6, 2024.
13. Goyal, S., Rallabandi, B., Gattupalli, K. K., & Medarametla, M. S. (2025). Digital twin-enabled context-aware networks: Enhancing smart grid resilience through predictive intelligence. In 2025 2nd International Conference on Recent Trends in Electrical, Electronics and Computing Technologies (ICRTEECT) (pp. 1–6). IEEE. https://doi.org/10.1109/ICRTEECT67512.2025.11448880
14. Khan, S. (2019). Sales force security compliance: An in-depth study of GDPR, HIPAA, and PCI-DSS enforcement in cloud-based CRM systems and their implications for global enterprises. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, 8(9), 2211–2219. https://doi.org/10.15662/IJAREEIE.2019.0809010
15. S. Choudhary, S. Kumar, S. Gowroju, M. Gulhane, and R. Sri Lakshmi, Eds., Genomics at the Nexus of AI, Computer Vision, and Machine Learning. Hoboken, NJ, USA: John Wiley & Sons, 2024.
16. Kalal, M., & Tanner, O. C. (2025). Autonomous exception management in SAP S/4HANA manufacturing through multi-agent generative AI and event-driven supply networks. International Journal of Core Engineering & Management, 8(4), 130–137.
17. Jaiswal, I. A. (2023). Intelligent Cybersecurity Framework for Large-Scale RESTful Service Architectures. International Journal of Research Radicals in Multidisciplinary Fields, ISSN: 2960-043X, 2(1), 178–184. Retrieved from https://www.researchradicals.com/index.php/rr/article/view, 252.
18. Rallabandi, B. R. (2018). Joint deployment and operational energy optimization in heterogeneous cellular networks under traffic variability. International Journal of Communication Networks and Information Security, 10(3), 638–647. https://ijcnis.org/index.php/ijcnis/article/view/8604
19. S. Choudhary, S. Kumar, M. Gulhane, and M. Kumar, "Secured automated certificate creation based on multimodal biometric verification," in Multimodal Biometric and Machine Learning Technologies: Applications for Computer Vision, pp. 269–281, 2023.
20. N. Pachauri, V. Suresh, M. V. V. Prasad Kantipudi, R. Alkanhel, and H. A. Abdallah, "Multi-Drug Scheduling for Chemotherapy Using Fractional Order Internal Model Controller," Mathematics, vol. 11, no. 8, Art. no. 1779, 2023.
21. Singh, M., Rallabandi, B., Gattupalli, K. K., & Sai Medarametla, M. (2025). Autonomous private 5G field area networks: Achieving secure, scalable, and self-healing smart grid communications. In 2025 2nd International Conference on Recent Trends in Electrical, Electronics and Computing Technologies (ICRTEECT) (pp. 1–6). IEEE. https://doi.org/10.1109/ICRTEECT67512.2025.11448712
22. Tripathi, N., Gattupalli, K. K., Rallabandi, B., & Medarametla, M. S. (2025). AI-native Cloud-RAN orchestration for enterprise private 5G using digital twin models. In 2025 1st IEEE Uttar Pradesh Section Women in Engineering International Conference on Electrical Electronics and Computer Engineering (UPWIECON) (pp. 851–857). IEEE. https://doi.org/10.1109/UPWIECON67212.2025.11390348
23. Rallabandi, B. R. (2020). MEC-native 5G systems: Orchestration algorithms for ultra-low latency cloud-edge integration. International Journal of Intelligent Systems and Applications in Engineering, 8(4), 398–408. https://ijisae.org/index.php/IJISAE/article/view/7889
24. Desai, A. B., Rallabandi, B., Gattupalli, K. K., & Medarametla, M. S. (2025). Agentic AI for rural connectivity and spectrum-aware networks to power smart agriculture ecosystems. In 2025 2nd International Conference on Recent Trends in Electrical, Electronics and Computing Technologies (ICRTEECT) (pp. 1–6). IEEE. https://doi.org/10.1109/ICRTEECT67512.2025.11448879
25. C. H. Neetha and M. P. Kantipudi, "Adaptive Edge-Based Bilinear Interpolation for Smart Healthcare," in IET Conference Proceedings CP824, vol. 2022, no. 26, Stevenage, U.K.: The Institution of Engineering and Technology (IET), 2022, pp. 7–13.

26. H. K. Jani, M. V. V. Prasad Kantipudi, G. Nagababu, D. Prajapati, and S. S. Kachhwaha, "Simultaneity of Wind and Solar Energy: A Spatio-Temporal Analysis to Delineate the Plausible Regions to Harness," Sustainable Energy Technologies and Assessments, vol. 53, Art. no. 102665, 2022.
27. M. Kalal, Y. R. Krishna, B. Jayswal, B. Konduru and V. S. A. Kondru, "Metaheuristic Optimization-Driven Information Management System for Enterprise Intelligence," 2026 IEEE 15th International Conference on Communication Systems and Network Technologies (CSNT), Al-Khobar, Saudi Arabia, 2026, pp. 1417–1423, doi: 10.1109/CSNT69054.2026.11502494.
28. Cherladine, K., Rallabandi, B. R., Goel, A., Kaushik, K., Dhillon, A., & Soni, M. (2025). Generative adversarial defense models for simulated cyberattack scenarios in virtual networks. In 2025 International Conference on Digital Innovations for Sustainable Solutions (ICDISS) (pp. 1–6). IEEE. https://doi.org/10.1109/ICDISS68238.2025.11320686
29. S. Rani, D. Ghai, and S. Kumar, "Reconstruction of wire frame model of complex images using syntactic pattern recognition," in IET Conference Proceedings CP791, vol. 2021, no. 11, pp. 8–13, 2021.
30. Goyal, S., Gattupalli, K. K., Rallabandi, B., & Medarametla, M. S. (2025). Integrating terrestrial and non-terrestrial networks for reliable rural coverage in agriculture and smart grids. In 2025 1st IEEE Uttar Pradesh Section Women in Engineering International Conference on Electrical Electronics and Computer Engineering (UPWIECON) (pp. 846–850). IEEE. https://doi.org/10.1109/UPWIECON67212.2025.11390331
31. R. Aluvalu, R. Venkatachalam, V. Uma Maheswari, R. J. Anandhi, M. V. V. Kantipudi, and S. Prashanth Mallellu, "IWHO-Based Cluster Head Selection for Vehicle-to-Vehicle Communication in Intelligent Transport System," International Journal of Transport Development and Integration, vol. 8, no. 2, pp. 154–166, 2024.
32. Jaiswal, I. A. (2024). AI-powered observability and incident prediction in distributed enterprise platforms. Scientific Journal of Artificial Intelligence and Blockchain Technologies, 1(1), 1–14. https://doi.org/10.63345/sjaibt.v1.i1.201
33. Rana, M., Srinivas, S., Jamili, L. K., Jaiswal, I. A., Nakka, S., & Kasetti, S. (2025, May). Real-Time Monitoring and Prediction of Blood Sugar Levels in Diabetic Patients with Functional Models. In 2025 International Conference on Engineering, Technology & Management (ICETM) (pp. 1–6). IEEE.
34. Swathi, S. Kumar, S. Rani, A. Jain, and R. K. M. V. N. M., "Emotion classification using feature extraction of facial expression," in Proceedings of the 2022 2nd International Conference on Technological Advancements in Computational Sciences (ICTACS), pp. 283–288, 2022.
35. S. Kumar, S. Choudhary, S. Gowroju, and A. Bhola, "Convolutional neural network approach for multimodal biometric recognition system for banking sector on fusion of face and finger," in Multimodal Biometric and Machine Learning Technologies: Applications for Computer Vision, pp. 251–267, 2023.
36. M. V. V. Prasad Kantipudi, S. Vemuri, N. S. Pradeep Kumar, S. Sreenath Kashyap, and S. Eslamian, "Proposing Model for Water Quality Analysis Based on Hyperspectral Remote Sensor Data," in Handbook of Hydroinformatics, Elsevier, 2023, pp. 317–324.
37. M. S. Prashanth, R. Aluvalu, and M. V. V. Kantipudi, "Enhancing Health Product Traceability on the Blockchain: A Novel Approach for Supply Chain Management Inspection to AI," EAI Endorsed Transactions on Pervasive Health & Technology, vol. 10, no. 1, 2024.
38. Khan, S. (2018). The role of zero-trust models in database security: Eliminating implicit trust and enforcing continuous verification in enterprise data access systems. International Journal of Research in Electronics and Computer Engineering, 6(1), 1622–1628.
39. S. Velamuri, S. K. Sudabattula, M. V. V. Prasad Kantipudi, and N. Prabaharan, "Q-Learning Based Commercial Electric Vehicles Scheduling in a Renewable Energy Dominant Distribution Systems," Electric Power Components and Systems, pp. 1–14, 2023.
40. V. Saini, A. Jain, Anurag, and M. V. V. Prasad Kantipudi, "An Efficient Approach for Improving the Performance of Autonomous Vehicle Using Advanced Computer Vision," in International Conference on Soft Computing and Pattern Recognition, Cham, Switzerland: Springer Nature Switzerland, 2023, pp. 164–174.