The ISC Int'l Journal of Information Security
ISeCure

October 2025, Volume 17, Number 3 (pp. 1–12)
http://www.isecure-journal.org

Presented at the ISCISC'2025 in TEHRAN, Iran.

Decentralised Plagiarism Detection System for Open Textual Educational Resources based on Blockchain Technology **

Sina Fattahi Ardakani [1], and Maedeh Mosharraf [1,*]
[1] *Department of Computer Science and Engineering, Shahid Beheshti University of Tehran, Iran*

ARTICLE INFO.

Keywords:
Open Educational Resources, Plagiarism, Blockchain, LSH Algorithm, Text Similarity Detection

Type:

doi:

ABSTRACT

Open Educational Resources (OER) have become a valuable tool for expanding access to quality education. However, managing intellectual property rights in OER environments remains a challenge, particularly in verifying content authenticity and protecting creators' rights. To address this challenge, the study proposes a decentralised approach to copyright management for OER. Our solution processes OER textual content using defined windows and the Locality Sensitive Hashing (LSH) algorithm to detect exact and partial similarities efficiently. By integrating blockchain technology and the InterPlanetary File System (IPFS), we establish a transparent, decentralised platform for storing and managing resources. To assess its effectiveness, the proposed system was implemented and tested on a dataset of 2,600 OER articles. The evaluation demonstrated perfect performance, with 100 per cent precision and recall across both direct and paraphrased plagiarism detection test sets. The results indicate that this technological integration can serve as a robust foundation for enhancing transparency and protecting authors' rights within the OER ecosystem.

1 Introduction

In recent years, advances in information and communication technologies have profoundly transformed the methods of teaching and learning. Today, access to digital resources and scientific information has become a fundamental necessity for the growth and development of societies. One of the most prominent manifestations of this transformation is the emergence and expansion of Open Educational Resources (OER), which have established a foundation for the free, affordable, and widespread sharing of knowledge [1]. These resources broaden learning opportunities and enhance the knowledge lifecycle. Yet, their increased use requires addressing challenges in intellectual property, authenticity, and security.

Major challenges include the difficulty of identifying ownership rights [2, 3], the inability of copyright holders to control the usage of their digital works effectively [2, 4–7], issues arising from the vast volume of online content and complex legal processes [2], threats to the information security and vital interests of original creators [2, 8, 9], unaddressed problems stemming from legal conflicts in cases of international online copyright infringement [5], and proving con-

* Corresponding author.
**The ISCISC'2025 program committee effort is highly acknowledged for reviewing this paper.
Email addresses: s.fattahiardakani@mail.sbu.ac.ir, m_mosharraf@sbu.ac.ir
ISSN: 2008-2045

tent authenticity [10]. If these issues are not properly resolved, creators' motivation to produce and share educational resources will decline [11]. Addressing these challenges directly contributes to the sustainability of OER [12].
Traditionally, these challenges have been addressed through a centralised server that stores and manages data and educational resources. However, such reliance on a single server can create vulnerabilities and limit scalability. Such a system allows users to access the resources they need; however, despite certain advantages, this method also presents important challenges [10]:

- **Single point of failure:** Any issue with the central server can result in the entire system becoming inaccessible.
- **Inability to track changes:** The system makes it difficult to accurately and transparently track and document modifications and updates.
- **Content authenticity issues:** The absence of effective mechanisms for verifying the accuracy and authenticity of resources undermines confidence in resource quality.
- **Copyright and intellectual property challenges:** Centralised systems often lack efficient frameworks for protecting creators' rights and properly managing intellectual property.
- **Sustainability concerns:** Long-term storage of data on a central server may lead to access issues and potential data loss.

Blockchain technology, as an innovative solution, can effectively address these challenges. The main features of blockchain—including transparency, security, and decentralised traceability—make it a suitable tool for ensuring copyright protection and safeguarding intellectual property rights in OER. Blockchain can automate processes such as content validation, ownership determination, and real-time tracking of modifications, thereby preventing unauthorised alterations and misuse of content [13].
This study aims to use blockchain technology to verify the authenticity of educational content and ensure copyright compliance during its distribution. The decentralised and transparent nature of blockchain provides a reliable foundation for managing and sharing Open Educational Resources. The specific objectives include: (1) identifying blockchain-based methods for protecting the copyright in OER, (2) enhancing system performance in recording transactions related to educational content on the blockchain, (3) developing approaches to reduce the costs of registering and generating transactions for OER on the blockchain, (4) designing a system to measure the similarity of content, and (5) preserving copyright and preventing plagiarism even in small parts of a resource.
The remainder of this article is organised as follows. Section 2 reviews related work and provides a comprehensive categorisation of existing approaches to copyright management in OER. Section 3 details the proposed methodology, practical implementation of the system, and technical specifications. Section 4 presents the results of evaluations conducted in various plagiarism scenarios. Section 5 analyses the strengths and weaknesses of the proposed system in comparison with other approaches. Finally, Section 6 presents the conclusions and offers suggestions for future research.

2 Related Work

Previous research on using blockchain to protect copyright in OER has mainly focused on two areas: employing the InterPlanetary File System (IPFS) for decentralised storage and applying watermarking techniques to secure content. However, these methods do not fully resolve issues like scalability and trust among stakeholders. Accordingly, prior work can be categorised into four categories:

2.1 Using neither IPFS nor Watermarking

This category has focused on the pure use of blockchain for protecting the copyright of OER. For example, [2] has presented a system for digital copyright protection whose main innovation is the use of a licensing platform to create a controlled environment and integration of smart contracts to automate processes. In [8], a blockchain system with hidden blocks and digital fingerprinting has been proposed, which has two parts: off-chain for authentication and on-chain for licensing. The key innovation lies in solving storage space and privacy issues and using a weighted consensus algorithm to improve efficiency. Furthermore, [14] has proposed a blockchain-based e-book transaction system that enables direct e-book purchases between authors and readers and uses smart contracts to manage rights and payments.

2.2 Using Watermarking without IPFS

The second category of research has used watermarking techniques for content protection. [4] has proposed a blockchain-based digital watermarking system in which cryptographic watermarks are embedded in content and related information is recorded on the blockchain. In [15], a secure and usable watermarking system between users has been proposed using blockchain technology and smart contracts, which enables tracking and controlling content distribution.

2.3 Using IPFS without Watermarking

The third research group has focused on combining IPFS with blockchain to improve scalability. [3] has presented a copyright management platform with a two-layer architecture that employs a distributed ledger, IPFS-blockchain integration and smart contracts for transaction automation. In [5], a decentralised system based on a consortium blockchain and a proof-of-authority algorithm was introduced. This system stores metadata on the blockchain and employs round-robin scheduling for leader selection and international coordination. [10] has created an open educational resources management framework by combining Ethereum and IPFS to eliminate single points of failure and reduce storage burden. Additionally, [9] has also provided a personal information ownership verification method with elliptic curve cryptography, zero-knowledge proof for privacy preservation, and distributed storage to solve capacity limitations.

2.4 Using both IPFS and Watermarking

The fourth category represents the most complex approach, combining IPFS, blockchain, and watermarking. [6] has presented a digital rights management system with two APIs for metadata, integration of external storage with blockchain, and a multi-signature tracking mechanism. [7] also introduced an image protection framework that combines a zero-watermarking algorithm, the Ethereum blockchain, and IPFS, which is used for off-chain storage.

2.5 Comparison of Previous Works

Table 1 consolidates the advantages and limitations of the four categories discussed in the previous subsections, supported by specific studies.

In Category 1, approaches using neither IPFS nor watermarking [2, 8, 14] integrate blockchain with smart contracts to automate the digital rights lifecycle and enable robust ownership verification. These methods achieve high consensus throughput [8] and strong tamper-resistance [2, 14] while addressing privacy concerns [8]. However, all three works operate at the level of complete content objects and cannot detect partial reuse or modified fragments. Moreover, the on-chain storage of encrypted content [14] and the growth of large-scale networks [8] create significant storage requirements.
Category 2 solutions, employing watermarking without IPFS [4, 15], enhance deterrence by rewarding informants [4] and embedding robust cryptographic watermarks resistant to countermeasures [15]. Both enable authenticity checks through blockchain-recorded watermark metadata. Nevertheless, like Category 1, they cannot trace partial copying, and their blockchain-only storage approach risks excessive network and storage overhead [4, 15].
Category 3, IPFS without watermarking [3, 5, 9, 10], shifts large content storage off-chain for permanence and scalability, while retaining blockchain for metadata integrity and smart contract governance. Systems in [3] and [10] ensure decentralised and immutable storage via IPFS, [5] introduces proof of authority consensus for efficient international coordination, and [9] uses cryptography and zero-knowledge proofs to protect privacy. Despite these merits, inability to detect partial plagiarism persists, and designs such as [5] introduce partial centralisation through reliance on consortium notaries.
Finally, Category 4, combining IPFS and watermarking [6, 7], offers the most comprehensive design. By placing large content in IPFS and storing ownership assertions in blockchain, these systems remove third-party dependence [6, 7], improve privacy control [6], and reduce computational demand [7]. The use of zero-watermarking [7] and multi-signature tracking [6] strengthens protection against unauthorised distribution. However, applicability is largely confined to specific media formats, particularly images and videos, and detection of partial plagiarism remains unsupported.
In sum, across all four categories in Table 1, the prevailing limitation lies in their inability to identify partial or subtly modified copies of OER. This shortcoming justifies the problem addressed in this study, as such undetected reuse undermines proper attribution, weakens intellectual property protection, and reduces trust in open sharing ecosystems. One possible solution is to integrate fine-grained similarity detection algorithms with decentralised storage and immutable ownership records, thereby improving content authenticity and copyright protection. To this end, the present work combines a Locality Sensitive Hashing (LSH) based text similarity module with IPFS and consortium blockchain infrastructure, enabling precise detection of both exact and partial plagiarism while preserving scalability, transparency, and security.

3 Methodology and Implementation

In this study, we present a decentralised and reliable system for detecting plagiarism and protecting intellectual property rights, with a particular emphasis on open textual educational resources. The system integrates three key components: the LSH algorithm for text similarity detection, the IPFS network for distributed data storage, and blockchain technology for guaranteeing data integrity and ownership validation. This architecture is designed to detect even subtle instances of plagiarism in textual content. At

Table 1. Comparison of Previous Blockchain-Based OER Protection Methods

Category	Advantages	Disadvantages
Using neither IPFS nor Watermarking	• Automated / integrated digital rights management • Effective rights protection • Addresses privacy challenges • High consensus speed • Capability to verify ownership • Data tamper-resistance	• Cannot detect partial copying • High storage requirements as network grows
Using Watermarking without IPFS	• Possibility of rewarding informants • Resistant to offenders' countermeasures	• Cannot detect partial copying • High network and storage demand
Using IPFS without Watermarking	• Supports copyright management with smart contracts • Addresses international challenges and allows pre-sharing review • Decentralised and resilient infrastructure • Ensures data integrity, immutability, and security	• Cannot detect partial copying • Relative centralisation in some designs
Using both IPFS and Watermarking	• Secure licensing and reduced computational costs • Eliminates need for third parties • Enhanced security and privacy • More robust watermarking algorithm	• Cannot detect partial copying • Limited to specific content types (image/video)

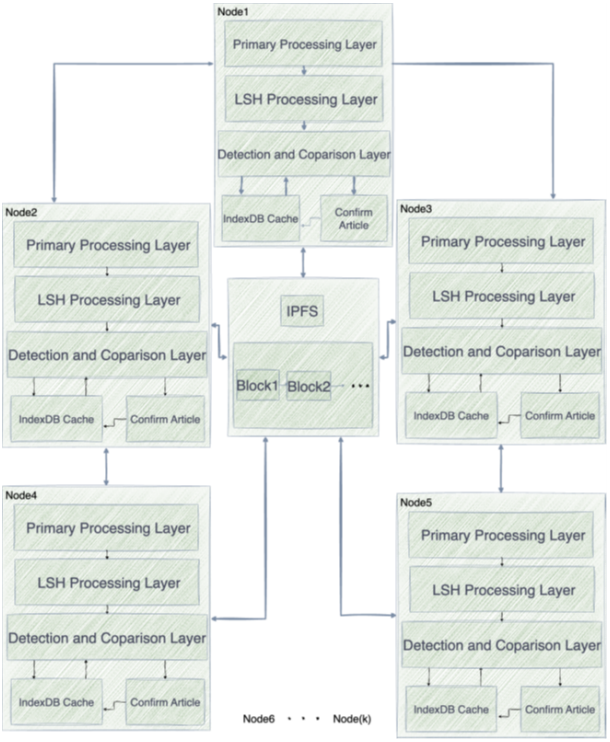

Figure 1. A Blockchain-Based System Infrastructure for Protecting the Copyright of Textual Educational Content

the same time, it ensures scalability, transparency, security, and resilience against censorship, thereby providing a robust foundation for trustworthy copyright management in educational environments. Figure 1 illustrates an overview of the proposed system infrastructure.

In this system, each node represents an independent publisher operating within the open-access ecosystem. These publishers participate in a consortium blockchain network, where all nodes collectively contribute to recording and validating new submissions. Consensus over article registration is achieved through a permissioned mechanism, ensuring that only authorised publishing entities can add records while still maintaining transparency and tamper-resistance. The consortium blockchain offers clear advantages over a fully public blockchain: transaction costs are significantly lower, consensus is faster due to a smaller network, and governance can be tailored to the needs of participating publishers. By restricting validation rights to trusted entities, the system achieves both efficiency and security while preserving the transparency essential to open-access collaboration.

This system utilises a combination of modern Web3 technologies and efficient hashing algorithms. Its architecture consists of four main components, each playing a specific role in the overall system performance:

- **Data collection:** This component gathers scientific articles from some digital repositories. It processes a collection of open-access articles by converting them from HTML to plain text.
- **Text processing and similarity detection:** The preprocessing process is performed using Natural Language Toolkit (NLTK) and custom algorithms. The main operations include sentence segmentation and N-gram shingle generation, with results stored in Indexed-DB as a local cache. Subsequently, using the LSH algorithm with Min-Hash and the text-repetition-ratio calculation system, the content is either approved or rejected.
- **Multi-layer storage system:** A hybrid architecture of IPFS and blockchain is proposed, in which bulk data is stored in IPFS, while metadata along with CIDs are maintained in smart contracts.
- **User interface:** A dynamic web interface provides users with the ability to upload files, monitor their processing status, and view results. This interface securely connects to a blockchain network to facilitate transactions and data interactions.

The details of each component are described below.

Table 2. List of Notations

Notations	Definition
n	Total number of hash functions used in the LSH algorithm
b	Number of bands in the banding technique
r	Number of rows per band ($r = n/b$)
s	LSH similarity threshold
k	Number of hash functions applied in Min-Hash computation
t	Number of bands into which the hash vector is divided (alternative notation for b)
Repetition ratio	Formula (1), ratio of duplicate sentences (weighted) to total sentences
CID	Content Identifier in IPFS
PMC API	Europe PubMed Central Application Programming Interface (data source)
IPFS	InterPlanetary File System (distributed storage)
LSH	Locality Sensitive Hashing (similarity detection algorithm)
Min-Hash	Signature method for estimating Jaccard similarity
UI	User Interface of the plagiarism detection system
N-gram	Sequence of N consecutive items from text (word or character level)
Jaccard similarity	Measure of similarity between finite sets

3.1 Data Collection

This component serves as the system input and is responsible for sourcing scientific content from reliable sources. Given that open-access articles are a primary form of OER, they were selected for evaluation in this research. For this purpose, the Europe PMC API service was used to search and retrieve open-access articles, ultimately collecting 2600 valid scientific articles. The process includes extracting article identifiers, downloading complete HTML content, and converting it to plain text format.
The system automatically manages the article list and performs initial data preparation. Network error management and content quality control are applied at this stage to ensure that only healthy and processable articles proceed to subsequent stages. This approach ensures that the input database maintains optimal quality.

3.2 Text Processing and Similarity Detection

This component is considered the core of the system and includes three separate subsystems, which are described below. Table 2 provides a consolidated list of all symbols, parameters, and abbreviations used in this section.

3.2.1 Sentence Extraction

In this research, the NLTK library has been used for text segmentation into sentences. NLTK, using the Punkt algorithm and custom settings for scientific abbreviations (such as "et al.", "Fig.", "p."), provides the capability for accurate detection of sentence boundaries in scientific texts.

3.2.2 Repetition Ratio Computation

One of the main innovations of this system is its Repetition Ratio Computation methodology for detecting the degree of content duplication. This system considers not only the number of duplicate windows, but also analyses the pattern of their distribution. If the repetition ratio computed by this system falls below the specified threshold, the content is classified as inappropriate and will not be registered.

The repetition ratio formula operates based on consecutive sequences of duplicate windows. In this method, long sequences of repetitive windows (which indicate direct copying) receive much higher ratios than scattered repetitive windows (which may be coincidental).

This system uses the following approach:

- For n consecutive repetitive sentences: negative score $= 3^{n-1}$
- For separate repetitive sentences: negative score $= 1$

It should be noted that negative scores are directly related to the repetition ratio. After calculating the negative scores for the entire content, the sum of scores is divided by the total number of content windows to obtain the repetition ratio as defined in Equation 1.

$$RepetitionRatio = \frac{totalRatio}{totalSentences} \tag{1}$$

If this ratio exceeds the repetition ratio threshold, the content is rejected as inappropriate content and will not be registered in the system at all. This scoring method is optimal because long sequences of repetitive windows (which indicate direct copying) receive much higher ratios than scattered repetitive windows (which may be coincidental).

By conducting experiments on the initial dataset and examining the repetition ratio of each article, it was observed that the repetition ratio of articles is mostly in the range of 0 to 0.3. Considering that there is a possibility of common repetitive sentences being scattered in articles, and these articles do not contain scientific plagiarism, a threshold of 0.3 was considered. Table 3 shows the output of repetition ratios extracted from the examined articles.

3.2.3 Duplicate Detection through LSH Algorithm

This section introduces the LSH algorithm as the core of the detection framework. It outlines the hash generation process, the use of banding for efficient

Table 3. Repetition Ratio Output for Evaluated Articles

Ratio	Articles	Ratio	Articles	Ratio	Articles	Ratio	Articles
0.00	545	0.16	5	0.33	1	0.78	1
0.01	601	0.17	1	0.34	1	0.79	1
0.02	406	0.18	4	0.39	1	0.82	1
0.03	314	0.19	3	0.41	1	0.88	1
0.04	216	0.20	3	0.42	1	0.90	1
0.05	157	0.21	1	0.46	1	1.28	1
0.06	114	0.22	2	0.48	2	1.43	1
0.07	72	0.24	3	0.49	1	1.53	1
0.08	57	0.25	4	0.50	1	1.90	1
0.09	35	0.26	1	0.53	1	2.49	1
0.10	30	0.27	3	0.56	1	2.94	1
0.11	26	0.28	2	0.59	1	3.01	1
0.12	12	0.29	2	0.63	1	3.20	1
0.13	16	0.30	3	0.68	1	5.38	1
0.14	6	0.31	1	0.69	2	14.75	1
0.15	7	0.32	3	0.77	1	32.51	1

search, and Jaccard similarity for final comparison. The method for optimising parameters to balance precision, recall, and computational efficiency is also described.

3.2.3.1 Generate Hashes

The LSH algorithm operates on the principle that if two texts are similar in content, their hashes should also be similar. In essence, LSH generates a type of hash for each text that enables fast and efficient comparison [16]. To generate hashes for each OER, the following four steps are essential.

Step 1: Advanced shingle system

The first step in processing any text is to convert it into defined windows, which, in this study, are considered to be sentences. Each window is then further divided into smaller units known as shingles.

To improve detection accuracy and cover different types of similarity, several types of shingles have been implemented in the system, each playing a specific role in the similarity analysis process:

(1) **Individual words:** After filtering stop words, which included 48 commonly used stop words, and words shorter than three characters that provide little information in similarity detection, the remaining words are considered as single-word shingles. This level of shingles enables the detection of common vocabulary and general content of texts.
(2) **Word 2-grams:** For detecting similar phrases and common two-word combinations, 2-gram shingles are used. This type of shingle can preserve the relative order of words and identify specialised terms and meaningful phrases. All possible 2-grams are generated without exception to achieve maximum coverage.
(3) **Word 3-grams:** For longer sentences and more complex linguistic patterns, 3-gram shingles are employed. This level of analysis enables the detection of more complex syntactic structures and preservation of short sentence meanings. 3-gram generation is performed only for sentences with at least three words.
(4) **Character 5-grams:** For fuzzy matching and detection of minor spelling variations, character 5-gram shingles are used, which provide very good accuracy in detecting character patterns. This method gives the system the ability to resist obfuscation techniques such as spelling variations, typing errors, and substitution of similar characters. It is applied only to texts longer than 10 characters.

This multi-layered combination enables the detection of a wide range of similarities from direct copying to minor structural changes.

Step 2: Min-Hash calculation

After the shingles are generated, the next step is to calculate the min-hash. This process involves applying k different hash functions. Each hash function maps every shingle to a large numeric value. For each hash function, the minimum value among all shingles in a given window is selected.
The process works as follows: Suppose a window contains five shingles. Hash function number one maps these five shingles to the numerical values 123, 456, 789, 234, and 567. The smallest value, 123, is chosen as the first hash element. This procedure is repeated for the remaining k-1 hash functions, ultimately producing an array of k numbers (the hash vector).

Step 3: Banding technique for fast search

One of the key innovations of LSH is the banding technique. In this approach, the $k-dimensional$ hash vector of each window is divided into t bands, each consisting of $\frac{k}{t}$ elements. Each band is then converted into a single hash value. This allows for the search to be conducted over similar bands rather than across all hash vectors.
The underlying logic is as follows: if two windows are truly similar, the probability that at least one of their bands is identical is very high. Therefore, to identify similar windows, it suffices to search among those that share at least one band. This reduces the computational load from millions of comparisons to only a few hundred.

Step 4: Calculating jaccard similarity

Once candidate pairs with potential similarity have been identified through banding, the exact degree of

similarity must be determined. To achieve this, the Jaccard Similarity metric is employed, which calculates the ratio of the number of identical elements in two hash vectors to the total number of elements. If this value exceeds the LSH threshold, the two windows are considered similar.

3.2.3.2 Determining Optimal Parameters

The LSH algorithm has been implemented with optimal settings for duplicate content detection. To determine the optimal parameters, a systematic approach based on specific criteria and comprehensive evaluation was adopted, which ultimately led to the selection of 20 hash functions, 10 bands, and an LSH similarity threshold of 0.4.

The design of this system was based on three main criteria:

(1) **Elimination of false positives:** The system should not identify dissimilar texts as similar
(2) **Paraphrase detection:** The system should have the capability to detect structural similarities when content is rewritten
(3) **Computational efficiency:** Too many hash functions can increase computational complexity

Given that the first criterion is much more important, using a lower LSH threshold is necessary to detect various types of similarity. This is because when sentences are paraphrased, the similarity level that this algorithm achieves drops below 0.6 (based on tested outputs). To find the optimal parameter combination for this project, the following steps were taken:

Step 1: Parameter range definition

- **Number of hash functions:** The number of hash functions was evaluated by increasing the count in increments of 10 within the range of 20 to 50. Values below 20 were excluded, as they significantly increased the false positive ratio in our tests. Increasing beyond 50 yielded no noticeable improvement in detection accuracy while adding unnecessary computational complexity. Intermediate values within the tested range exhibited minimal performance variation; for example, results obtained at 15, 18, and 20 hash functions were nearly identical.
- **Similarity threshold:** The similarity threshold was examined over the interval [0.3, 0.7] in increments of 0.1. Thresholds below 0.3 produced a higher false positive ratio, while thresholds above 0.7 typically corresponded to nearly identical inputs, rendering further comparisons unnecessary. Similar to the hash function tests, intermediate values showed little difference.

Step 2: Calculating the optimal number of bands

To determine the optimal number of bands, the theoretical LSH relationship was used, which defines the approximate detection threshold [17]. Equation 2 shows the approximate ratio of LSH similarity threshold with the number of bands and the number of rows in each band.

$$s = (\frac{1}{b})^{\frac{1}{r}} \tag{2}$$

where:

- s is the similarity threshold for LSH
- b is the number of bands
- r is the number of rows in each band

Given the relationship n = r × b (where n is the total number of hash functions), the optimal number of bands can be calculated for any chosen number of hash functions.

Step 3: Intelligent combination generation algorithm

Considering the limitations mentioned above, an algorithm was developed to generate valid parameter combinations in accordance with LSH theory and empirical requirements. The process begins by selecting candidate values for the number of hash functions from the ranges defined in the preceding step, and likewise selecting target similarity thresholds from the same pre-established interval. For each chosen number of hash functions, all possible values of b and r are computed, keeping only those combinations that meet the logical constraints. The theoretical similarity threshold is then calculated for each valid (b, r) pair and compared with the candidate target thresholds. Only those parameter sets whose theoretical and target thresholds differ by at most 0.1 are retained for further evaluation.

Table 4 shows the algorithm's output for the parameter ranges defined.

Step 4: Parameter evaluation and selection

The evaluation criteria include these three metrics:

- **Precision:** Tested on the original dataset that was free of plagiarism. Equation 3 shows this criterion.
- **Recall:** Tested on samples that were manually paraphrased. Equation 4 shows this criterion.
- **F1-Score:** A balanced combination of the two above criteria, shown in Equation 5.

Table 4. Parameter Combinations

Number of Hash Functions	Hash Bands	Band Size	Similarity Threshold	Theoretical Threshold
20	4	5	0.7	0.758
20	5	4	0.6	0.669
20	5	4	0.7	0.669
20	10	2	0.3	0.316
20	10	2	0.4	0.316
30	8	4	0.5	0.595
30	8	4	0.6	0.595
30	16	2	0.3	0.250
40	8	5	0.6	0.660
40	8	5	0.7	0.660
40	10	4	0.5	0.562
40	10	4	0.6	0.562
40	20	2	0.3	0.224
50	10	5	0.6	0.631
50	10	5	0.7	0.631

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

$$F1Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{5}$$

After testing all combinations and sorting results based on F1-Score, as well as considering the third criterion of minimising the number of hash functions, we calculated the final scores using Equation 6:

$$Score = 0.6 \times F1Score + \frac{1}{n} \tag{6}$$

The optimal combination was determined as follows:

- Number of hash functions: 20
- Number of bands: 10
- Similarity threshold: 0.4

The obtained combination has several key advantages, which include:

- **High precision:** No false positives were observed in the original dataset.
- **Optimal paraphrase detection:** The 0.4 threshold enables the detection of minor changes in sentence structure.
- **Computational balance:** 20 hash functions create an appropriate balance between accuracy and speed.

By substituting b=10 and r=2 in Equation 2, the value s = 0.316 is obtained, which is the theoretical threshold for this combination. The theoretical threshold has an acceptable proximity to the selected threshold of 0.4 and provides the possibility of finer tuning for paraphrase identification.
This scientific methodology ensures that the selected parameters are optimised not only based on empirical experimentation but also on LSH theoretical principles.

3.3 Multi-layer Storage System

One of the prominent features of this project is the implementation of a multi-layer storage architecture, in which each layer serves a specific function and operates in coordination with the other layers.

Layer 1: Local cache with Indexed-DB

The most fundamental storage layer is the local cache, which is maintained by each blockchain node. This cache comprises three main components:

(1) The bands section stores all calculated bands for rapid search operations. It is organised as a tree structure, where the first key is the band number, the second key is the band's hash, and the value is a list of windows sharing the same band hash.
(2) The sentences section establishes a direct mapping between the hash of each window and its original text. This component is used for fast retrieval of the original window text.
(3) The min-hash section holds the min-hash vector for each window. These hash vectors are used to compute similarity with new windows.

Using Indexed-DB for this cache offers several advantages. First, the data is persistently stored on the local device, ensuring that it remains available across system restarts. Second, it supports storing large volumes of data. Third, it enables highly efficient read and write operations.

Layer 2: IPFS for distributed storage

IPFS is a distributed network that identifies files based on their content rather than their location. This means that each file receives a unique fingerprint, and if the content is altered, the fingerprint changes accordingly [18]. This property ensures complete data integrity. In this system, instead of storing all information directly on the blockchain, only the hashes of each unique content window are collectively stored in IPFS, which in turn generates a unique CID for them. Utilising IPFS offers several significant advantages. First, the storage cost is considerably lower than that of blockchain. Second, data can be accessed from different locations around the world. Third, it provides resistance to censorship and to failures of centralised servers.

Layer 3: Blockchain for data integrity

The top layer of the storage system is the blockchain, which stores only the CIDs generated by IPFS. This is a highly efficient design choice, as blockchain space is limited and expensive, but it provides unparalleled security and integrity guarantees. In this research, only one CID per content item is stored in the smart contract. The CID serves as a key to access the complete data in IPFS. Since CIDs are short strings, the storage cost on the blockchain is both minimal and acceptable.

Information retrieval(reverse process):

The information retrieval system is designed to provide free and transparent access to all data. Any user can easily obtain a complete list of content available in the system. The retrieval process is carried out in three stages. First, users can retrieve all CIDs stored in the blockchain as a list by calling public functions of the smart contract. This operation is completely public and unrestricted, providing transparent access to the entire system content. The second stage involves retrieving complete data from IPFS. With the CIDs in hand, users can retrieve each CID individually from the IPFS network. This data includes an array of hashes of unique windows for each content, stored in JSON format. This stage enables the retrieval of small portions of each source separately. The final stage is retrieving the original text through hash mapping. For this purpose, a local cache structure is used that maps each sentence hash to its original text.

In this method, during initial processing, each unique sentence is stored in the cache along with its computed hash. During retrieval, using the hashes stored in IPFS, the original text of sentences is extracted from the local cache. This method preserves data security and integrity while ensuring rapid access to original content. Figure 2 shows the sequence of operations performed by a node in this blockchain.

3.4 User Interface

The user interface, developed using React.js, acts as a bridge facilitating interaction between users and the sophisticated backend system. Through this interface, users can process articles individually or in batches, track the progress of similarity detection, and examine detailed results—including similarity percentages and matched text segments. The interface also includes an advanced statistical panel that enables the display of detected repetition ratio distributions, number of processed articles, number of rejected articles, and overall system performance. Additional features include searching and displaying stored sentences of each article through IPFS CIDs, network error management, blockchain transaction status display, and the ability to reset cache and statistics.

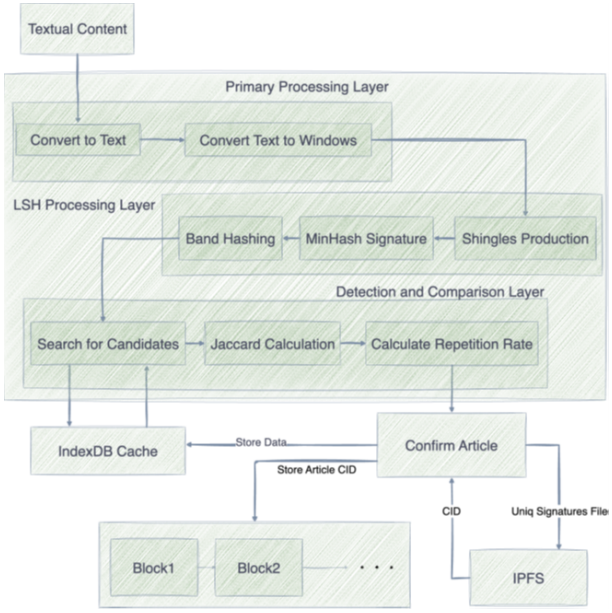

Figure 2. The Workflow of Each Node in the Blockchain Network

4 System Evaluation

For comprehensive system evaluation, 2300 out of the 2600 articles in the dataset were allocated for training and configuring the LSH system, creating the hash vector database, and forming the hash band structure. The remaining 300 articles — the test set — were divided evenly into three plagiarism categories to evaluate detection performance under different scenarios.

To generate plagiarism cases within the test set, we applied two approaches:

- **Approach 1:** 30 per cent of the article content (based on article size) was modified, and each plagiarism window was inserted separately into the original article. The 30 per cent threshold was chosen based on the LSH similarity threshold of 0.3 obtained in earlier experiments — smaller changes would substantially increase the false positive ratio.
- **Approach 2:** 5 per cent of the article content (based on article size) was modified, with all plagiarism windows placed consecutively in the article. Due to the exponential accumulation of negative scores, this 5 per cent is sufficient for the similarity ratio to exceed 0.3.

We will now examine the three categories of plagiarism:

- **Direct plagiarism:** This type comprises 100 test-set articles in which every plagiarism window was copied exactly, with no changes, from the original test-set content. Plagiarism was

generated twice — once using Approach 1 and once using Approach 2— and both approaches produced identical results.
The purpose of this test is to evaluate the system's ability to detect the simplest and most obvious form of plagiarism.

- **Paraphrased plagiarism:** This type comprises 100 test-set articles in which every plagiarism window was rewritten through large language models. The rewriting involved changes in word order and minor structural modifications. Plagiarism was generated twice — once using Approach 1 and once using Approach 2 — with all modified windows sourced directly from the original test-set articles. Both approaches produced identical results. This test examines the system's capability to detect plagiarism involving only superficial changes.
- **Semantic plagiarism:** This type comprises 100 test-set articles in which every plagiarism window underwent complete semantic rewriting by large language models, yet the main concept and idea were preserved. In this plagiarism type, the words, sentence structure, and even writing style changed, while the scientific and conceptual content remained essentially the same.
Plagiarism was generated using only Approach 2, with all rewritten windows derived from the original test-set articles, because our methodology struggles to detect semantic changes and cannot identify this type of plagiarism through exponential score accumulation. This represents the most challenging detection scenario for any plagiarism detection system.

4.1 Detailed Results and Performance Analysis

The following part details the system's detection results for the three plagiarism categories in the test set:

- **Direct plagiarism detection performance:** The results of the first category test demonstrate flawless system performance in detecting direct copying. All 100 articles were identified with a 100 per cent detection ratio, and no false negatives were observed. Detection speed was also very satisfactory in this category, with an average processing time of less than 2 seconds per article.These results confirm that the LSH algorithm is highly efficient for detecting direct lexical similarity.
- **Resistance against paraphrasing:** In the second category test, the system also demonstrated remarkable performance. Despite structural changes and the use of synonyms, all 100 paraphrased articles were detected with 100 per cent accuracy. This result indicates that the combination of different types of shingles and proper adjustment of the LSH similarity threshold has enabled the system to be resistant against superficial changes.
- **Semantic detection challenge:** The third category test revealed the main limitation of the system. Out of 100 articles with semantic plagiarism, less than 2 per cent were detected by the system. This limitation stems from the LSH algorithm's dependence on lexical matching and demonstrates that the system is unable to understand deep semantic relationships. In cases where detection occurred, it was mainly due to the preservation of some key phrases or specialised terminology.

Table 5. Precision, Recall, and F1-score for Three Plagiarism Types with Weighted Averages

Category	Precision	Recall	F1-Score
Direct Plagiarism	100%	100%	100%
Paraphrased Plagiarism	100%	100%	100%
Semantic Plagiarism	100%	2%	4%
Weighted Average	100%	67%	80%

4.2 Comprehensive Statistical Analysis

A detailed evaluation was performed across three plagiarism categories: direct, paraphrased, and semantic. The results are summarised in Table 5.

Precision achieved 100 per cent across all categories, meaning no false positives were detected—critical for safeguarding original works from erroneous plagiarism flags. Recall was 100 per cent for direct and paraphrased plagiarism but dropped to only 2 per cent for semantic plagiarism, reflecting the difficulty of capturing deep semantic similarities with the current implementation. The weighted F1 score reached 80.48 per cent, an acceptable balance given the priority on precision over recall in applications where protecting original content integrity outweighs exhaustive semantic detection.

4.3 Test Environment and System Specifications

The experiments were conducted on a MacBook Pro M1 Pro with a 10-core ARM64 processor, 32 GB of unified memory, and a 1 TB SSD, operating under macOS Monterey. The implementation was developed using Node.js 18.17.0, React 18.2.0, Hardhat for local blockchain simulation, and js ipfs 0.62.3. Ethereum was selected as the primary platform for

local testing due to its mature ecosystem, extensive documentation, and comprehensive simulation tools, which enabled accurate emulation of network conditions and smart contract behaviour without incurring transaction fees. The compatibility of the Ethereum Virtual Machine (EVM) ensured that contracts executed in the local environment could be deployed consistently on both the Ethereum mainnet and any EVM-compatible chain.

While the Ethereum mainnet offers strong security and a proven consensus mechanism, its high transaction costs make it less suitable for large-scale, cost-sensitive deployments. For higher scalability, the production version can be deployed on a consortium blockchain built atop EVM-compatible Layer 2 solutions (e.g., Polygon, Arbitrum, Optimism). This approach can combine the efficiency of Layer 2 networks with permissioned governance by trusted publishing entities, preserving decentralisation and transparency while ensuring performance and cost effectiveness for open yet controlled academic publishing.

5 Discussion

Analysis of the proposed system's methodology and performance indicates notable strengths and weaknesses, which are summarised in Table 6.

6 Conclusion

In this study, we present a decentralised copyright management solution for OER that integrates the LSH algorithm, blockchain, and IPFS. This approach enables accurate text similarity detection without relying on centralised infrastructure. The system operates at granular levels within a secure, transparent, and scalable multi-layered architecture. Although evaluated on open access articles, it can be applied to any textual content. Its main limitation lies in detecting semantic plagiarism involving deep paraphrasing, due to reliance on lexical similarity. Future directions include the development of semantic-aware plagiarism detection methods to better handle meaning-preserving rewrites. In practice, this could extend to industrial applications such as consortium blockchains among publishers and academic platforms, ensuring interoperability, cost-efficiency, and collaborative governance in copyright protection.

References

[1] Maedeh Mosharraf and Fattaneh Taghiyareh. The role of open educational resources in the elearning movement. *Knowledge Management & E-Learning*, 8(1):10, 2016.

[2] Yanhui Liu, Jianbiao Zhang, Shupei Wu, and Muhammad Salman Pathan. Research on digital copyright protection based on the hyperledger fabric blockchain network technology. *PeerJ Computer Science*, 7:e709, 2021.

[3] Yi Ouyang, Xianghan Zheng, Xiaoliang Lu, Lin Xiaowei, and Shengyin Zhang. Copyright protection application based on blockchain technology. In *2019 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*, pages 1271–1274. IEEE, 2019.

[4] Bo Zhao, Liming Fang, Hanyi Zhang, Chunpeng Ge, Weizhi Meng, Liang Liu, and Chunhua Su. Y-dwms: A digital watermark management system based on smart contracts. *Sensors*, 19(14):3091, 2019.

[5] Md Mainul Islam and Hoh Peter In. Decentralized global copyright system based on consortium blockchain with proof of authority. *IEEE Access*, 11:43101–43115, 2023.

[6] Zhaofeng Ma, Ming Jiang, Hongmin Gao, and Zhen Wang. Blockchain for digital rights management. *Future Generation Computer Systems*, 89:746–764, 2018.

[7] Baowei Wang, Shi Jiawei, Weishen Wang, and Peng Zhao. Image copyright protection based on blockchain and zero-watermark. *IEEE Transactions on Network Science and Engineering*, 9 (4):2188–2199, 2022.

[8] Gabin Heo, Dana Yang, Inshil Doh, and Kijoon Chae. Efficient and secure blockchain system for digital content trading. *IEEE Access*, 9:77438–77450, 2021.

[9] Shaoqi Yuan, Wenzhong Yang, Xiaodan Tian, and Wenjie Tang. A blockchain-based privacy preserving intellectual property authentication method. *Symmetry*, 16(5):622, 2024.

[10] Ujjal Marjit and Prabhakar Kumar. Towards a decentralized and distributed framework for open educational resources based on ipfs and blockchain. In *2020 International Conference on Computer Science, Engineering and Applications (ICCSEA)*, pages 1–6. IEEE, 2020.

[11] Susan D'Antoni. Open educational resources: Reviewing initiatives and issues, 2009.

[12] Sanjaya Mishra. Open educational resources: Removing barriers from within. *Distance education*, 38(3):369–380, 2017.

[13] Severin Bonnet and Frank Teuteberg. Impact of blockchain and distributed ledger technology for the management of the intellectual property life cycle: A multiple case study analysis. *Computers in Industry*, 144:103789, 2023.

[14] Jeonghee Chi, Jangyeon Lee, Nakyung Kim, Jeewoo Choi, and Soyoung Park. Secure and reliable blockchain-based ebook transaction system

Table 6. Advantages and Limitations of the Proposed System

Advantages of the Proposed System	Limitations of the Proposed System
• **Independence from centralised infrastructure:** Using the distributed architecture of IPFS and blockchain, unlike traditional systems dependent on central servers • **Very high scalability:** Ability to process very large volumes of data without performance degradation • **Privacy preservation:** Not storing complete content and using cryptographic hashes • **Exceptional speed:** Having O(n) time complexity in search compared to direct methods with exponential complexities • **Censorship resistance:** Impossibility of data deletion or modification by central authorities • **Complete transparency:** Ability for all users to verify all transactions and changes • **Partial plagiarism detection:** The capability to identify copying of small portions of content	• **Weak semantic detection:** Limited capability in detecting semantic plagiarism compared to methods based on natural language processing and deep models • **Parameter dependency:** Need for precise tuning of LSH parameters for each type of content • **Limitation in deep change detection:** Inability to detect complete rewrites while preserving meaning • **Technical knowledge requirement:** Implementation and maintenance complexity of the system • **Network dependency:** Need for continuous connection to IPFS and blockchain networks • **Transaction costs:** Despite optimisation, still requiring gas fee payments for blockchain transactions • **No offline support:** Inability to function without internet access

for self-published ebook trading. *PloS one*, 15 (2):e0228418, 2020.

[15] Franco Frattolillo. Blockchain and smart contracts for digital copyright protection. *Future Internet*, 16(5):169, 2024.

[16] Aristides Gionis, Piotr Indyk, Rajeev Motwani, et al. Similarity search in high dimensions via hashing. In *Vldb*, volume 99, pages 518–529, 1999.

[17] Anand Rajaraman and Jeffrey D Ullman. *Mining of massive datasets*. Autoedicion, 2011.

[18] Juan Benet. Ipfs-content addressed, versioned, p2p file system. *arXiv preprint arXiv:1407.3561*, 2014.

Sina Fattahi is a Master's student in E-Commerce at Shahid Beheshti University, Tehran, Iran, since 2023. He earned his B.Sc. degree in Computer Engineering from Isfahan University of Technology in 2023. His research interests include blockchain technologies, cryptocurrencies, and plagiarism detection systems, reflecting his dedication to advancing secure and innovative digital solutions.

Maedeh Mosharraf is an Assistant Professor of Computer Engineering Software and Information Systems at Shahid Beheshti University where she has served since 2021. She received her M.Sc. (2013) and Ph.D. (2019) in Computer Engineering from University of Tehran. Her current research involves Digital transformation, Electronic commerce, and Blockchain technologies and cryptocurrency.