systems

MDPI

Article

DOCS: A Data Ownership Confirmation Scheme for Distributed Data Trading

Yang Liu [1], Yang Zhang [1], Yongsheng Yang [1] and Yan Ma [2,3,*]

[1] Institute of Logistics Science and Engineering, Shanghai Maritime University, Shanghai 200120, China
[2] School of Accounting, Nanjing University of Finance and Economics, Nanjing 210023, China
[3] School of Computing, National University of Singapore, Singapore 119077, Singapore
* Correspondence: yanma@nufe.edu.cn

Abstract: Data assets trading can encourage owners to distribute data and achieve large-scale data aggregation to promote the development of the supply chain system. Blockchain is a promising platform for constructing a decentralized data marketplace. The data may face risks in the marketplace, such as illegal theft, malicious tampering, or illegal distribution in the transactions process. The data ownership confirmation in a blockchain-empowered marketplace has attracted much attention in recent years. However, challenges still remain, including maintaining data integrity, traceability of illegal data, and accountability. In this paper, we propose a new data ownership confirmation scheme (DOCS) in the transaction scenario of blockchain-empowered distributed data assets trading. It integrates smart contracts, data-embedding technology, and data fingerprint to realize ownership confirmation and protection of data assets in transactions. DOCS ensures reliable mapping between on-chain data ownership information and off-chain data entities, which assists with the accurate prosecution of the illegal distribution of data assets. We demonstrate that DOCS can have desirable security properties in multiple attack models.

Keywords: supply chain; data ownership; data asset trading; block chain; smart contract

Citation: Liu, Y.; Zhang, Y.; Yang, Y.; Ma, Y. DOCS: A Data Ownership Confirmation Scheme for Distributed Data Trading. *Systems* **2022**, *10*, 226. https://doi.org/10.3390/systems10060226

Academic Editors: Anders Hansen Henten and Iwona Windekilde

Received: 30 September 2022
Accepted: 14 November 2022
Published: 19 November 2022

1. Introduction

The effective application of data helps enterprise supply chains achieve business process transformation and product and service innovation, and it helps to improve supply chain operations. Enterprises wanting to maintain their competitive advantages must pay attention to the application of big data, which requires extensive access to data from internal and external sources, and data trading across organizations and between chains becomes very important, and data trading becomes an important means to strengthen data resource integration, open information silos, and activate data assets. However, data trading faces risks in practice, such as unclear ownership, complicated authorization, lack of transparency of transactions, and privacy leakage. Figure 1 shows a general data trading scenario. The data user sends a data purchase request, and the data owner responds to the request. When the data user pays, the data owner embeds the watermark into the data and sends it to the data user. However, the virtual, non-exclusive, and lossless characteristics make data easy to be tampered with, resold, leaked, and used beyond the scope in the process of circulation. The characteristics of the zero marginal production cost and the difficulty of complete physical delivery make it impossible for the ownership, use, and control of data to be delivered uniformly. Therefore, the static data watermarking model cannot be applied to dynamic data market transactions.

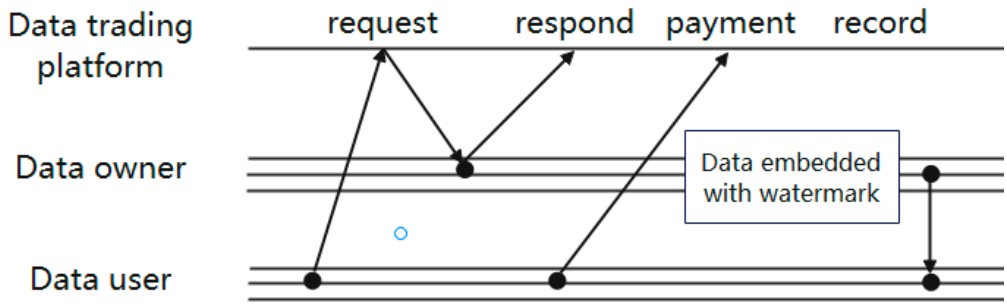


Figure 1. Data transaction scenario.

Blockchain, first proposed by Satoshi Nakamoto [1], is a public ledger, maintained by decentralized nodes for the distributed sharing and storage of data. Blockchain has the characteristics of decentralization, anonymity, privacy, traceability, and tamper resistance, which has attracted great attention from academia and industry (such as supply chain, Internet of things, and medical fields) [2]. New concepts, such as smart contracts [3] and smart attributes that originated from blockchain technology, were quickly accepted by the economic market. A smart contract is a computer transaction protocol that enforces the terms of contract, allows trusted transactions without third parties, and ensures that those transactions are traceable and irreversible. In recent years, blockchain technology has been successfully applied to IoT platforms [4,5], medical data sharing systems [6], data privacy protection [7], supply chains [8–10], biomedical research [11], and financial transactions [12]. The decentralization of the blockchain paves the way for data transactions in the supply chain data marketplace.

Related work. Zhao, Y. et al. [13] proposed a new protocol for distributed data transactions that uses ring signatures to enhance the privacy of data provider identities. In a ring signature, a user selects a group of users, called the ring, to generate a signature, where the verifier can be confident that the signature was generated by a member of the ring but cannot reveal which person actually generated the signature. The protocol also extends double-authentication prevention signatures (DAPS) to penalize signers who generate two signatures for messages with the same title and different payloads, and this guarantees the fairness of transactions between data providers and data consumers. Xiang, Y. et al. [14] proposed a smart-contract-based data trading scheme. The scheme uses smart contracts to ensure fairness of data sharing and data copyright in transactions and minimizes the risk of partial/combined resale or leakage of data by using a multi-type-based watermarking strategy. Jing, N. et al. [15] proposed a blockchain-based code copyright management system. The original verification model of code based on abstract syntax tree is applied to the verification process of blockchain to realize the copyright verification and protection of original code. However, there is a problem of originality verification cost and verifier's dilemma [16]. Xu, Y. et al. [17] proposed a game theory based Nash equilibrium model between watermarking robustness and data quality. The model uses a secure hashing algorithm to establish the mapping relationship between data groups and watermark bits and uses an improved particle swarm optimization algorithm to solve the optimal solution for each data group's data variation under the data availability constraint and then modifies the data accordingly to complete the embedding of the watermark bits and protect the copyright of the data. Kumar, R. et al. [18] proposed a distributed image- and video-sharing platform based on IPFS (Interstellar File System). The platform detects copyright infringement of multimedia by calculating the similarity between perceptual hashes (pHash) stored in the blockchain. Nasonov, Denis. et al. [19] proposed a distributed big data platform in which a blockchain-based distributed digital data market is used to ensure the integrity of data transactions. Zhou, J. et al. [20] addresses the trade-off dilemma between the effectiveness of data retrieval and the leakage risk of data indexing in distributed data transactions, and they propose a framework for distributed data transactions (DDV) by combining data embedding and similarity learning. The framework

uses a privacy-preserving data-embedding procedure as an input to measure the similarity between data entries and achieves effective retrieval in data transactions while preserving data privacy. Elias Strehle and Martin Maurer [21] proposed the DibiChain protocol for the discovery and exchange of supply chain information, which is built on top of a distributed data store that maintains a high degree of anonymity and unlinkability while ensuring a high degree of privacy by minimizing data in the shared data store, avoiding persistent user identifiers and communicating anonymously with minimal intermediaries. Nawaz, A. et al. [22] proposed EdgeBoT, a platform for IoT based on smart contracts, considering the potential changes in interaction topology in data transaction scenarios. EdgeBoT enables more diverse interaction topologies between nodes in the network and external services, enabling direct data transactions at edge devices while guaranteeing data ownership and end-user privacy.

However, most of the current research on data ownership confirmation in data trading is focused on improving digital watermarking technology and similarity detection. This can only cover the detection of illegal data and cannot fundamentally cover the accurate tracing and timely accountability of illegal data. The current trading platform construction has no standard system for data ownership verification, traceability, and accountability.

Our contributions. This paper proposes a data ownership confirmation scheme (DOCS) for distributed data asset trading of the supply chain system, which has a credible and accountable architecture. We have studied in detail the structural methods of data storage, traceability, and accountability. (1) We adopt data signatures and similarity learning to enhance the reliable mapping between on-chain data ownership and off-chain data entities. It can effectively maintain the integrity of off-chain data. (2) We propose a smart contract-based data fingerprint generation protocol, which contains a two-part structure of mutual identity verification and data fingerprint generation. This ensures channel security under anonymous transaction networks and also achieves accurate traceability and market tracking of illegal data transactions. (3) We design a market supervision mechanism empowered by smart contracts to encourage market users to assist in prosecuting illegal data transaction in a timely manner.

The rest of this paper is structured as follows. Section 2 introduces the basic applications of DOCS, including data signatures, similarity learning, and smart contracts. Section 3 describes the structure of DOCS and the workflow and defines common data tenure attack models. Section 4 provides a security analysis of DOCS and demonstrates that DOCS can resist attacks on data tenure in data transactions. Section 5 evaluates the encoding performance and decoding performance of data-embedding techniques with supply chain data, and the experimental results show that data signatures can be used as reliable credentials for data ownership confirmation. Section 6 provides a conclusion.

2. Preliminary

2.1. Data Signature

Simple hash-based data signatures are unique and random. For the same data, as long as its content is slightly modified and finely distinguished from the original data, a new hash signature can be obtained even if the data does not satisfy the originality condition, and the characteristic relationship with the original data hash signature cannot be detected. In addition, the hash signature is irreversible and cannot be restored to the original data. When a data dispute arises, the hash signature cannot be used as the basis for the judge's decision.

Advances in deep learning have led to highly nonlinear embedding techniques, such as autoencoder [23], and recurrent neural networks for embedding in time series data [24]. The advantage of using data signatures based on a data-embedding technique instead of simple hash-based signatures is that it helps to achieve efficient data retrieval and similarity detection. The goal of data embedding is to project the data input into a generally lower-dimensional subspace so that the data input can be represented by a low-dimensional vector. As shown in Figure 2, the data-embedding framework consists of two major modules: the

encoding process and the decoding process. The input sample X is mapped to the feature space Z through the encoder (f), which is the encoding process; then the abstract feature Z is mapped back to the original space through the decoder (g) to obtain the reconstructed sample $\widetilde{X}$, which is the decoding process.

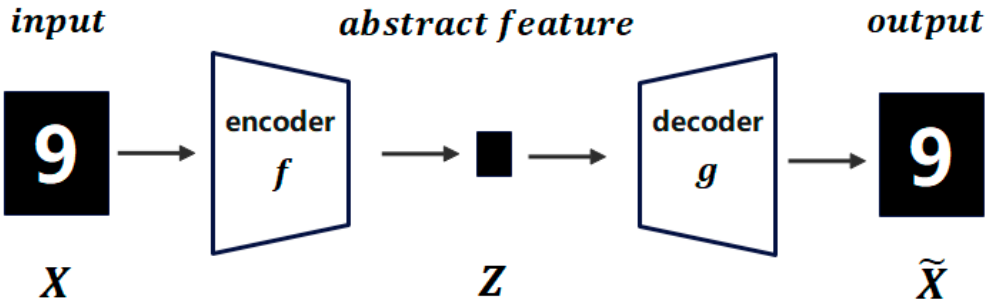


Figure 2. Data encoding and decoding process.

The optimization objective is to optimize the encoder and decoder at the same time by minimizing the reconstruction error, so as to learn the abstract feature representation Z for the sample input X.

$$f, g = argmin_{f,g} \ell(X_i,\ g_\gamma(f_\theta(X_i))) \tag{1}$$

where f is the embedding function, g is the inverse function, θ is the parameter of the f, and γ is the parameter of the g. The process of target optimization is the optimization process of θ and γ. Equation (1) is a process of objective optimization. Through it, we can obtain encoders and decoders with better performance.

$$f_\theta(X_i) = Z_i \tag{2}$$

When the data owner needs to update the data matrix or more data entities are available, data embedding can be done in a data-driven manner, so that we learn an embedding function such that the learned subspace can maximize the data recovery entity. Let $D = \{X_1, X_2 \cdots, X_n\}$ be a set of n data entities available for training; the data-driven embedding is given by the following objective function:

$$\min_{\theta,\gamma} \sum_{i=1}^{n} \ell(X_i, g_\gamma(f_\theta(X_i))) \tag{3}$$

where $\ell(\cdot,\cdot)$ is a loss function for evaluating the recovery error. Let θ^*, γ^* be the optimal solution in the pair (3); we can update the functional form of θ^* and f by calling the initialization contract.

$$V_j = f_{\theta^*}(X_j) \tag{4}$$

The embedding vector V_j is the data signature of the data matrix X_j.

In order to alleviate the problem of easy overfitting of the classic autoencoder, one way is to add random noise to the input layer of the traditional autoencoder to enhance the robustness of the model [23]. Another way is to combine the idea of regularization, by adding the autoencoder's Jacobian matrix paradigm to the autoencoder objective function to constrain the autoencoder to learn abstract features with anti-interference [25].

2.2. Similarity Learning

Distance metric learning has been extensively studied for decades and is widely used in computer vision, information retrieval systems, and bioinformatics [26]. It can greatly improve the performance of classification, clustering, and retrieval tasks [27]. Distance metric learning involves learning the distance relationship of specified data from pairs of similar but different points. In an information retrieval system, we can define similarity on specified data and learn a distance function d for efficient retrieval.

Let $(Z_1, Z_2) \in R_m$ represent two data abstract features, where m is the number of features. The Mahalanobis distance [28] between (Z_1, Z_2) can be expressed as:

$$d_M(Z_1, Z_2) = \sqrt{\| Z_1 - Z_2 \|_M^2} = \sqrt{(Z_1 - Z_2)^T M (Z_1 - Z_2)} \quad (5)$$

where M is the parameter matrix of the distance metric. We use distance metric learning to compute data similarity and make decisions. The metric learning algorithm is extended to multi-task settings when there are many tasks in [28]. In the multi-task setting, the distance for task t is defined as:

$$d_t(Z_1, Z_2) = \sqrt{(Z_1 - Z_2)^T (M_0 - M_T)(Z_1 - Z_2)} \quad (6)$$

The specific regularization term is defined as:

$$min_{M_0} = \gamma_0 \| M_0 - I \|_F^2 + \sum_{t=1}^{T} \gamma_t \| M_t \|_F^2 \quad (7)$$

The parameter γ_t controls the regularization of $M_t, t \in [0, \cdots, T]$. Multi-task metric learning can effectively improve the performance of distance metric for retrieval task learning.

2.3. Smart Contract

A smart contract is a self-executing program code first proposed by Nick Szabo [29]. Smart contracts are derived from the Bitcoin scripting language, a stack-based language that is not yet fully completed. Ethereum [30] is an alternative cryptocurrency for building the next generation of distributed applications that support smart contracts. Ethereum smart contract provides a more expressive and complete language, as well as the most widely reliable language; the transaction network in DOCS is built on Blockchain Ethereum, specifically; the basic functionality of DOCS is implemented through key smart contracts.

Some computationally intensive contracts (CICs) are very expensive to execute, which makes it impossible for us to execute complex algorithms with low gas cost. The implementation of CICs will also lead to the validator's dilemma problem [16], a miner must normally start mining a new block on one received only after verifying all its transactions. If the time taken to verify the transactions in the block is nontrivial then it delays the start of the mining process, thereby reducing the chances of the miner creating the next block. Skipping the verification step will save time but at the risk of quality. Although selecting a small group of miners to execute contract computations can reduce costs, it does not guarantee the trustworthiness of executing contracts.

YODA [31] proposes a method to implement CICs in the system while guaranteeing a threat model that allows Byzantine and selfish nodes in the system. YODA [31] selects one or more execution sets (ES) via Sortition to execute a particular CIC off-chain.

3. DOCS

3.1. DOCS Overview

In the DOCS, Blockchain Ethereum serves as the underlying blockchain infrastructure to build the transaction network in DOCS, where a combination of smart contract features can be enabled. There are also three participants: data owner, data user, and market users.

Blockchain Ethereum: Blockchain Ethereum is an open source public blockchain with smart contract functionality. The data owner and data user trade data on the Blockchain Ethereum.

Data owner: The data owner is usually the producer of the data. They have a list of topics to advertise the sales data, register the publication data with the Blockchain Ethereum, and generate a topic transaction.

Data user: The data user is usually a buyer of data. They query the data list through the Blockchain Ethereum and generate payment transactions to purchase the data.

Market users: Market users are data users who collect and trade data through black market. They are rewarded for assisting with the prosecuting of data users for illegal transactions.

The workflow of DOCS is shown in Figure 3. The data owner will then publish the list of topics on the Blockchain Ethereum. Data user search on the Blockchain Ethereum and request data on a specific topic to enter the publication stage. Data owner generate the data signature through the data-embedding function, and request to upload it to the transaction list. After the data user retrieves the availability of the data, the transaction enters the verification stage. The two parties conduct identity verification through the data fingerprint generation protocol based on smart contract, and after the verification passes, the transaction enters the payment stage; after the data user completes the payment, the subject data embedded in the data fingerprint is obtained, and the transaction enters the supervision stage; within the validity period of the supervision stage, market users obtain rewards by assisting with prosecuting data users for illegal transactions, and market users may also choose to resell data for profit.

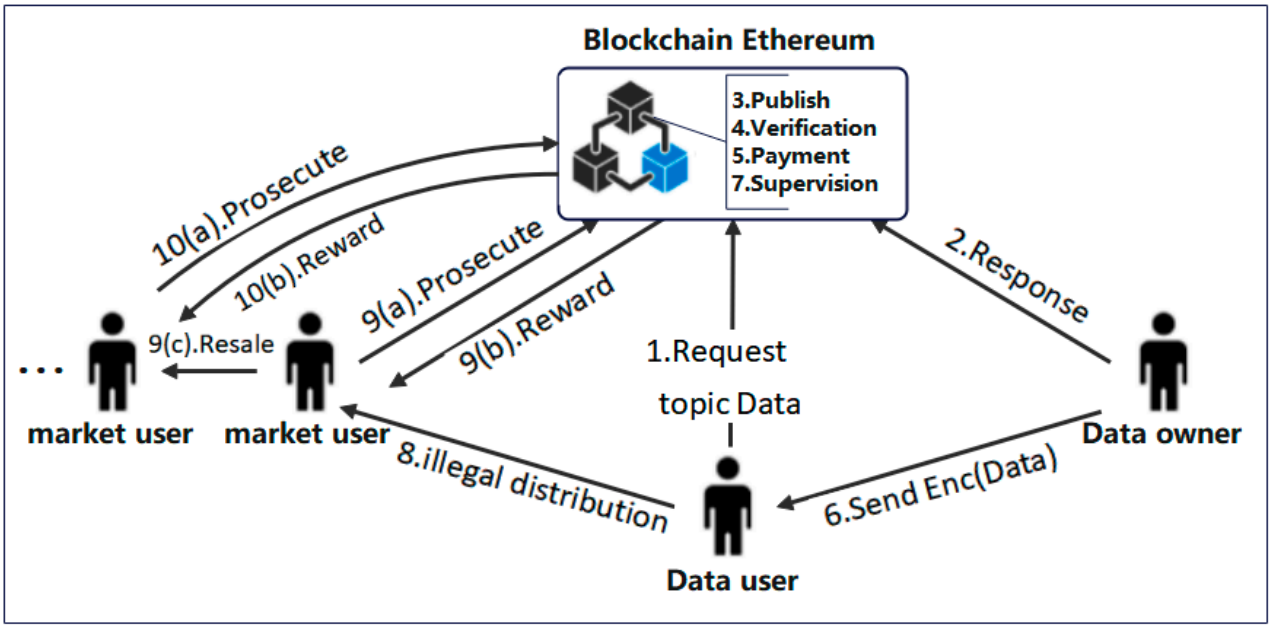


Figure 3. Architecture of DOCS. (a–c) indicates the sequence of market user' actions.

Adversary models. According to the real process of data trading, we identify six typical potential adversaries in distributed data transaction scenarios, which are the most common attacks on data ownership and the most threatening in terms of data transaction systems attacks. The specific definitions of these attacks are as follows:

Definition 1. *False identity attack. During the transaction process, the counterparty uses a false identity to evade the tracking of data fingerprint.*

Definition 2. *Repeat confirmation attack. After the adversary obtains the data copy of the data owner, it slightly modifies the data copy to obtain a new data ownership certificate, which is confirmed and traded on the chain.*

Definition 3. *Data corruption attack. Data are often stored and processed with the risk of data corruption, such as data loss and data distortion.*

Definition 4. *Illegal distribution. After the adversary obtains the data copy of the data owner, it circumvents the on-chain transaction network and conducts anonymous transactions off-chain.*

Definition 5. *Shared key attack. The adversary gets access to the data owner's encrypted data and causes data leakage by sharing the data decryption key.*

Definition 6. *Transaction fraud. In a transaction, the buyer and seller do not stay synchronized in the process of payment and delivery, specifically one peer is spoofed by another peer, resulting in the loss of data or tokens.*

The key symbols used in data trading are presented in Table 1.

Table 1. Key symbols and corresponding descriptions.

Notations	Description
L	Data Transaction List
$H(\cdot)$	Secure hash function
K_1	Secure session key of DU
K_2	Secure session key of DO
DO	Data owner
DU	Data user
DF	Data Fingerprint
TL	Transaction Protection Period
M_O	Deposit for DO
M_U	Deposit for DU
α	Random parameter of DO
β	Random parameter of DU
$G(\cdot)$	Security irreversible function
$Cert_{DO}$	Security certificate of DO
$Cert_{DU}$	Security certificate of DU
Sig_{Data}	Data signature
$Sig_{CA}(\cdot)$	CA signature

3.2. Workflow of DOCS

Publish. We argue that the process of publishing data to the Blockchain Ethereum is the confirmation process of data ownership, and Sig_{Data} can be used as a valid proof of data ownership. Algorithm 1 describes the publishing process implemented with smart contract.

Step 1: The data owner uploads a data signature Sig_{Data} based on data-embedding technology and the miner updates it to the blockchain transaction list $L(Sig_{Data})$ after verifying its legitimacy. The data user retrieves $L(Sig_{Data})$ and moves to the transaction verification phase after verifying the availability of the data through similarity learning.

Algorithm 1: Contract_publish

Input: Sig_{Data}, Issure, contract_state
Output: $L(Sig_{Data})$, contract_state
1. if Sig_{Data} = true
2. Sig_{Data} update to L
3. renew $L(Sig_{Data})$
4. contract_state=verification
5. else
6. return an error

Verification. Before data user can pay, we need an identifiable data fingerprint. For data transaction scenarios, data fingerprinting protocols that rely on third parties do not support anonymity, and the leakage of fingerprint information will also create risks for transaction participants. DOCS rely on data fingerprints to trace user and owner identities, so the security and trustworthiness of fingerprints is very important for member management and accountability tracking. In DOCS, a necessary but not sufficient condition for the credibility of a data fingerprint is to verify the identity of the other party. We propose a data fingerprint generation protocol based on smart contracts. The framework of the protocol

is shown in Figure 4. The protocol requires mutual authentication of participant identity, confirmation of the identity of the sender, and channel security, and it then generates DF through $H(\cdot)$. The process is as follows:

(1) Authentication initialization

Step 2: The data owner and data user obtain their certificates through CA authentication. The certificate structure is as follows:

$$Cert_{DO} = \{PubK_O, L(Sig_{Data}), Issuer, Algorithm, Sig_{CA}(\cdot)\}$$

$$Cert_{DU} = \{PubK_U, Issuer, Algorithm, Sig_{CA}(\cdot)\}$$

The initialization smart contract generates random numbers α and β, the data owner computes Q_O, and the data user computes Q_U and uploads them to the smart contract along with the certificate.

$$Q_O = G(\alpha)$$

$$Q_U = G(\beta)$$

Step 3: The data user computes K_1 and uploads c_1 and c_2 to the smart contract; the data owner computes K_2 and uploads c_3 and c_4 to the smart contract.

$$\begin{gathered} K_1 = \beta Q_O = \beta G(\alpha) \\ c_1 = Enc_{PubK_O}(K_1) \\ c_2 = Enc_{K_1}(Cert_{DU}) \\ K_2 = \alpha Q_U = \alpha G(\beta) \\ c_3 = Enc_{PubK_U}(K_2) \\ c_4 = Enc_{K_2}(Cert_{DO}) \end{gathered}$$

(2) Session key authentication

Step 4: The data owner decrypts to get K_1 and $Cert^*_{DU}$. The data owner then sends $Enc_{PubK_U}(K_1)$ to the smart contract. The data user decrypts to get K_2 and $Cert^*_{DO}$. The data user then sends $Enc_{PubK_O}(K_2)$ to the smart contract.

(3) Identity verification

Step 5: The data owner gets K_2^*; the data user gets K_1^*.

$$K_2^* = Dec_{PriK_O}\left(Enc_{PubK_O}(K_2)\right)$$

$$K_1^* = Dec_{PriK_U}\left(Enc_{PubK_U}(K_1)\right)$$

(4) Generate data fingerprint

Step 6: Smart contract generates a DF by $H(\cdot)$.

$$DF = H(PubK_O \oplus Sig_{Date} \oplus PubK_U)$$

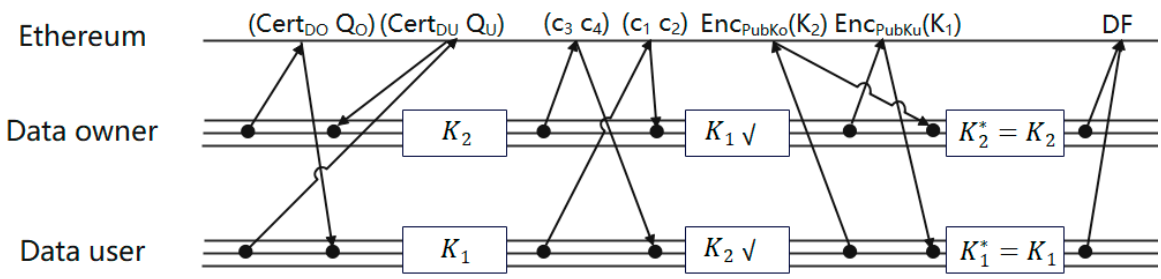


Figure 4. Schematic diagram of certificate-based data fingerprint generation.

Payment. Before the data user pays, both parties deposit a certain amount of deposit in the smart contract. If the payment is successful, the deposit will be returned after a time limit TL, and the transaction will enter the supervision stage. If payment fails, the data user will compensate the data owner for a certain loss, and the transaction will be terminated. Algorithm 2 describes the payment process implemented with smart contracts.

Step 7: The data user pays the data price, the data owner embeds DF to the corresponding subject data and uploads the encrypted data $Enc_{K_2}(Data)$ with DF to the cloud storage, and the data user downloads the decrypted data $Dec_{K_2}(Data)$.

Algorithm 2: Contract_payment

Input: M_O, M_U, TL, contract_state
Output: contract_state
1. if DU initiates a payment to DO
2. DO to embed DF in the data
3. DO sends $Enc_{K_1}(Data)$ to cloud storage
4. return M_O to DO after TL
5. return M_U to DU after TL
6. contract_state = supervision
7. else
8. termination transaction
9. destroy DF
10. return M_O to DO
11. DU compensates the loss from M_U to the DO

Supervision. There are two ways that market can obtain illegal copies of data. First, direct transactions between data user and market users. Second, transactions between market users. We greatly encourage all participants in the data market to assist with prosecuting a data user for unlawful conduct. When the market user purchases an illegal copy of the data, the smart contract will seek to upload the data fingerprint and send a verification request to the miner. If $DF^* = DF$ and the upload time t of DF^* is in an ownership protection period TL, return 1 to Contract_prosecution, then the market user sued successfully. If DF^* is invalid, or t exceeds one ownership protection period, then 0 is returned and the market user's prosecution fails. Algorithm 3 shows this process implemented with a smart contract. If return 1, smart contract Contract_payment will issue a reward b from M_U to the market user who successfully assist with prosecuting. Then, it will compensate v to the data owner, where v is the initial price of the data. Algorithm 4 describes the reward process implemented with smart contract.

Definition 7. *Reward mechanism. The price paid by market user u_i for a copy of the data is v_i. The reward for a successful prosecution is b. There is a scale factor λ_i for b and v for the price of the data, $b_i = \lambda_i v$ $(b_1 > b_2 > b_3 \cdots > b_n)$, and λ_i decreases in steps as the number of Contract_prosecution triggers increases.*

We assume that the data compensation is gradually reduced but not to zero, so that the deposit of data user always meets the requirements. The reason why this assumption can be made is that when the reward is low enough, the data owner has received enough compensation, and they are also satisfied that the data user will pay enough for the illegal distribution of data copies. Therefore, this game model is still valid.

Algorithm 3: Contract_prosecution

Input: DF^*, account(u_i)
Output: return 0 or 1
1. if $DF^* = DF$
2. $DF \rightarrow \langle PubK_O,\ PubK_U,\ \mathrm{L}(Sig_{Data})\rangle$
3. return 1 to contract_payment.rep(T)
4. else
5. return 0 to contract_payment.rep(T)
6. termination transaction

Algorithm 4: Expansion of Contract_payment

1. func rep( ):
2. var {account(u_i), T, b, t}
3. if (T = 1) $\wedge$ (t $\leq TL$)
4. successful prosecution
5. send b_i to account(u_i) from M_U
6. send v to DO
7. else if (T = 0) $\vee$ (t $> TL$)
8. prosecution failed

4. Security Analysis

In this section, we prove that DOCS can defend against various types of attacks on data ownership in data transaction scenarios.

Theorem 1. *DOCS can resist false identity attacks.*

Proof. There are two ways in which a data user can provide a false identity, which are analyzed as follows:

(1) The certificate itself is invalid. In DOCS, when the data owner receives $Cert_{DU}$, it will be verified by $PubK_U$ to $Sig_{CA}(\cdot)$. If the verification is successful, it means that $Cert_{DU}$ is valid. If the verification fails, it means that the data user holds an invalid certificate. Likewise, data user can be authenticated, $Cert_{DO}$, in the same way.

(2) Whether the data subject is the true owner of the certificate. Data user try to send other people's certificates to circumvent smart contract-based fingerprint generation protocols and avoid fingerprint tracking. The most effective way for DOCS to verify that the data subject is the true owner of the certificate is by verifying that the data subject actually owns the private key of $Cert_{DU}$. The data owner can obtain K_1, Q_U, and K_2^* of the data user during the transaction verification stage. From $K_1 = \beta G(\alpha)$, it can be reversibly deduced to β, and the data owner calculates $\mathrm{G}^*(\beta)$. If $\mathrm{G}^*(\beta) = Q_U = G(\beta)$ and $Cert_{DU}^* = Cert_{DU}$, it means that the data user's K_1 is correct. If $K_2^* = K_2$, the identity of the data user is correct. Likewise, data user can authenticate data owner in the same way. The data user can obtain K_2, Q_O, and K_1^* of the data owner during the transaction verification stage. From $K_2 = \alpha G(\beta)$, it can be reversibly deduced to α, and the data user can calculate G^*. If $\mathrm{G}^*(\alpha) = Q_O = G(\alpha)$ and $Cert_{DO}^* = Cert_{DO}$, it means that the data owner's K_2 is correct. If $K_1^* = K_1$, the identity of the data owner is correct.

Therefore, no matter how the adversary provides false identity information, it will be detected, and DOCS can resist false identity attacks. □

Theorem 2. *DOCS can defend against repeat confirmation attacks.*

Proof. After the data user purchases and obtains the data entity X, corresponding to $L(Sig_{Data})$, they attempt to slightly modify and reacquire a new data signature to upload to the blockchain network for ownership confirmation and to initiate a transaction. Data

signature based on data-embedding techniques are essentially abstract features of the data entity X, the data signature can be represented by a vector Z, equation (5) calculates the distance relationship between different data signatures, and the metric learning algorithm can be extended to a multi-task setup when there are many tasks (Equation (6)). In the following, we describe how this attack can be intercepted by a combination of data signature and similarity learning. □

We present YODA [31] in the defense process of DOCS to demonstrate that our defense is more robust. First, the anchor node broadcasts Sig_{Data}'s similarity learning request R to the entire network, and its retrieval scope includes the list of all serialized data signatures on the Blockchain Ethereum. Initialization smart contract pseudo-randomly selects miner M_i to join the execution set $ES = \{M_1, M_2, M_3, \cdots, M_i\}$ of R. M_i performs the similarity retrieval task of Sig_{Data} independently, returns the execution result $ER_i = \left(bool, R, Sig_{M_i}(\cdot), SR_i\right)$, and broadcasts it to other miners in ES, where bool represents the execution result of R is true or false; $Sig_{M_i}(\cdot)$ represents the signature of miner M_i, SR_i represents the result of RICE [31], and the miner who executes it through the PBFT consensus protocol reaches a consensus result ER. Then, the anchor node broadcasts ER to $\neg ES$, regenerates $ES^{'} \in \neg ES$ and re-executes R.$\neg ES$ maintains the result set $\{true, false, dispute\}$, where $true$ means X is original data, $false$ means X is duplicate data, $dispute$ means X is in dispute, and you need to submit it manually for verification. Finally, $\neg ES$ decides the final execution result from the result set through likelihood estimation; $\neg ES$ serializes the result and sends feedback to ES to terminate the computation. Therefore, the repeat confirmation attack of the data user can always be blocked by DOCS.

Theorem 3. *DOCS can defend against data corruption attacks*.

Proof. Data storage and processing are often accompanied by risks, such as data loss and data distortion. Since Sig_{Data} is reversible, the data owner can decode Sig_{Data} by decoder g and get X $= g_\gamma(Sig_{Data})$. Sig_{Data} is stored in the Blockchain Ethereum as an ownership credential, and X is permanently trusted due to the tamper-proof nature of the blockchain. The data owner can use X as the credential to audit the data entity under the chain and effectively maintain the integrity of the data entity under the chain. Therefore, DOCS can resist data corruption attacks. □

Theorem 4. *DOCS can defend against illegal distribution*.

Proof. The illegal distribution of data cannot be realized in our data transaction network based on Blockchain Ethereum because it will be blocked in the transaction response stage. Data users often choose to avoid on-chain transactions and resell copies of data on the black market. DOCS rely on credible data fingerprints DF and timely incentives to encourage market users to sue data users for illegal distribution in an anonymous network. We denote the set of market users who purchase data copies in the black market as U and $U = \{u_1, u_2, u_3, \ldots\ldots, u_i\}$, market users want to maximize their own profits no matter how they obtain data copies. The policy space of U is $(r, p, none) = (resale, prosecute, none)$, and the action set of U can be expressed as $\{r \wedge p, \neg r \wedge p, r \wedge \neg p, none\}$. We analyze the market users benefit matrix under the four actions, as shown in Table 2. □

Table 2. The benefits of market users under different actions.

	$r \wedge p$	$\neg r \wedge p$	$r \wedge \neg p$	*none*
u_1	$v_2 - v_1 + b_1$	$-v_1 + b_1$	$v_2 - v_1$	$-v_1$
u_2	$v_3 - v_2 + b_2$	$-v_2 + b_2$	$v_3 - v_2$	$-v_2$
u_3	$v_4 - v_3 + b_3$	$-v_3 + b_3$	$v_4 - v_3$	$-v_3$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
u_i	$v_{i+1} - v_i + b_i$	$-v_i + b_i$	$v_{i+1} - v_i$	$-v_i$

Starting from the row of Table 2, the user gets the greatest benefit when executing $r \wedge p$; starting from the column of Table 2, the user who sues the earliest can always get the greatest benefit. Therefore, the illegal distribution of the data user can always be traced back in time. In Table 3, we analyze the payoff matrix of market users, data owner, and data user in the case of action $r \wedge p$.

Table 3. Action is the payoff matrix of $r \wedge p$.

U	*DO*	*DU*
b_1	v	$-b_1 - v$
b_2	$2v$	$-b_2 - 2v$
b_3	$3v$	$-b_3 - 3v$
$\vdots$	$\vdots$	$\vdots$
b_i	iv	$-b_i - iv$

From Table 3, we can see that if the ith market user u_i successfully sues, the return to the u_i is b_i, the compensation to the data owner is iv, and the loss to the data user is $-b_i - iv$. Therefore, the sued data user will face huge compensation beyond the value of the data itself. Under this kind of game, data users will not distribute copies of data illegally on the black market. If data users choose to distribute copies of data illegally, data owners will not suffer losses.

In the supervision model, the earlier a market user sues, the more rewards they can receive. A market user has to sue before other market users in order to get higher re-wards, so this creates a competitive relationship between market users. In most cases, market users do not know the source of illegal data copies. We can rely on this com-petitive relationship to encourage market users to initiate timely assistance with pros-ecutions and improve the timeliness of the monitoring model. Although we cannot eradicate the continuous distribution of illegal data copies and need to rely on compe-tition to encourage market users to file lawsuits in a timely manner, we have established a game relationship between market users and data user in this way. If data users choose to illegally distribute copies of data to the black market, they are likely to face high penalties in a short space of time. Moreover, the penalties are much higher than the benefits obtained by illegally distributing data copies. In this game, data users are forced to remain rational.

Theorem 5. *DOCS can defend against shared key attack.*

Proof. In the DOCS, the data owner encrypts the data with a randomly generated temporary session key K_2. During the generation of K_2 by the data owner, the data owner can set permissions and policies so that K_2 can only be used within the specified scope of permissions. For example, K_2 will be invalid after data are decrypted by the data user. Or K_2 expires after a certain time limit has been exceeded. Therefore, data user cannot leak data through the shared key. □

Theorem 6. *DOCS can defend against transaction fraud.*

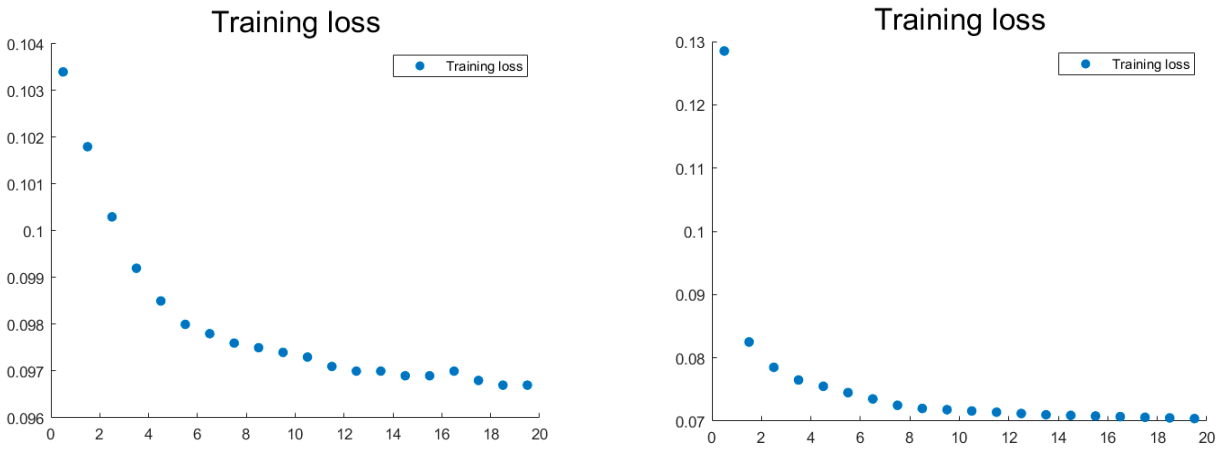
Proof. The data user attempts to refuse delivery of the data after the data user has paid. In this case, the data user may set a return value and a time limit in the payment phase of the smart contract to return a value that triggers the smart contract to take effect after receiving the complete data. If the data user does not return the data in time after receiving it, the smart contract automatically becomes effective after a time limit is exceeded and the data owner is paid.

The data user attempts to refuse payment after the data owner delivers the data. In DOCS, the data user refuses to send encrypted data in case the data owner refuses to pay, all of which does not happen. □

5. Performance Evaluation

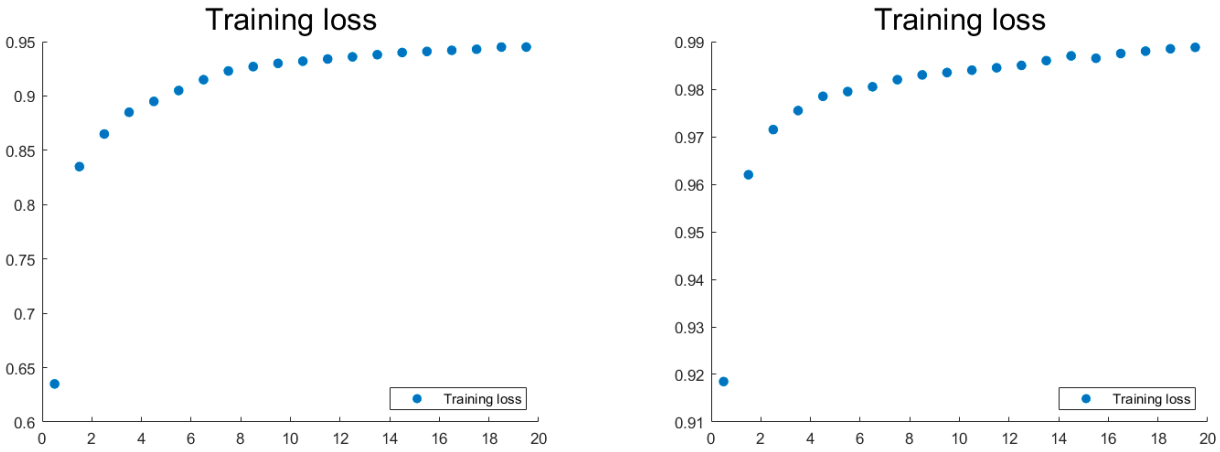
In this section, we evaluate the embedding performance and recovery performance of DOCS on real datasets. We use the supply chain geographic proximity data of nearly 30,000 listed enterprises as our simulation dataset, and the data information in the dataset contains user/supplier ID, spatial distance, and distance to user/supplier, etc. We can unify the data standards in the initializing smart contract and build a data standard repository chain network using Blockchain Ethereum, which is jointly constructed and maintained by all the nodes that join. After the new data standards are verified by the consensus algorithm of each node, they are linked to the standard library chain to ensure the stability and openness and transparency of the data standard library.

Our simulation platform is Intel(R) Core(TM) i5-5350U CPU @ 1.80GHz 8.00GB RAM and Windows 10 operating system. We evaluate the autoencoder-based stacked denoising autoencoder (SDAE) and convolutional autoencoder (CAE) performance in Figures 5 and 6, and the decoding efficiency in different dimensions in Table 4.



(**a**) (**b**)

Figure 5. The loss change of the self-supervised pre-training process of SDAE is shown in (**a**), and the loss of the self-supervised pre-training process of CAE is changed, as shown in (**b**).



(**a**) (**b**)

Figure 6. The variation in acc accuracy during SDAE unsupervised training is shown in (**a**), and the variation in acc accuracy during CAE supervised training is shown in (**b**).

Table 4. The decoding efficiency in different dimensions.

Dimension	Precision	Recall
500	0.771	0.650
1000	0.911	0.840
1500	0.919	0.878
2000	0.920	0.901
2500	0.922	0.916
3000	0.922	0.925
3500	0.923	0.932
4000	0.924	0.939
4500	0.925	0.941
5000	0.928	0.945
5500	0.927	0.943
6000	0.927	0.942

From Figure 5, it can be seen that the training loss of CAE self-supervision converges to about 0.07, which is smaller than 0.096 of SDAE. From Figure 6, the accuracy of CAE's supervised training process rises to 0.99, which is higher than SDAE's accuracy of 0.95. Therefore, CAE has certain advantages in reconstruction tasks and classification tasks in datasets.

Table 4 shows that the recovery performance of the decoder also tends to increase slowly as the data dimension increases, further illustrating that the recovery efficiency tends to saturate as the signature vector size increases. In Table 4, it can be seen that after 1000 dimensions, the improvement of recovery efficiency decreases significantly with the increase in embedding dimension. In order to achieve scalable feature representation and retrieval performance, we would like to use an embedding size that stays within a rational range, which compresses the raw data sufficiently without significantly sacrificing embedding accuracy. Therefore, we suggest using a 1000-dimensional embedding representation because it provides more than 30 times the compression of the original supply chain data while preserving most of the sparsity and temporal properties of the downstream tasks.

6. Conclusions

In this paper, we propose a distributed data ownership confirmation scheme (called DOCS) in a data transaction scenario. The advantage of a data transaction network built on Blockchain Ethereum is that it eliminates the single point of failure in the big data

market. We describe the data signature and fingerprint generation protocols in the DOCS architecture, as well as the market supervision mechanism empowered by smart contracts, and build a standard system for data ownership verification, traceability, and accountability, maintaining data integrity and enabling accurate traceability and timely accountability for illegal data transactions. We demonstrated that DOCS can resist different types of attacks. We analyzed the encoding performance and decoding performance of different autoencoders through supply chain data.

Most smart contract applications, including DOCS, face privacy concerns because of the conflict between privacy needs and the transparency of blockchains and smart contracts. In the Blockchain Ethereum, anyone can view the current state of the smart contract, which also contains information about personal consumption and more. An effective smart contract access control mechanism plays an important role in resolving the above conflicts, and our future research will be carried out on this basis.

Author Contributions: Conceptualization, Y.L. and Y.Z.; methodology, Y.L. and Y.M.; software, Y.Z.; validation, Y.L., Y.Z. and Y.M.; formal analysis, Y.L.; investigation, Y.L.; resources, Y.Z. and Y.L.; data curation, Y.Z.; writing—original draft preparation, Y.L. and Y.Z.; writing—review and editing, Y.L.; visualization, Y.Z.; supervision, Y.Y.; project administration, Y.Y. and Y.L.; funding acquisition, Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Singapore-UK Cyber Security of EPSRC under Grant Nos. EP/N020170/1, MOE Humanities, and Social Sciences Foundation of China under Grant Nos. 20YJCZH102.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. *Decentralized Bus. Rev.* **2008**, *21260*, 9.
2. Tschorsch, F.; Scheuermann, B. Bitcoin and beyond: A Technical Survey on Decentralized Digital Currencies. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 2084–2123. [CrossRef]
3. Sharma, P.; Jindal, R.; Borah, M.D. A Review of Smart Contract-Based Platforms, Applications, and Challenges. *Cluster Comput.* **2022**, *in press*. [CrossRef]
4. Wang, X.; Yu, G.; Zha, X.; Ni, W.; Liu, R.P.; Guo, Y.J.; Zheng, K.; Niu, X. Capacity of Blockchain Based Internet-of-Things: Testbed and Analysis. *Internet Things* **2019**, *8*, 100109. [CrossRef]
5. Leiba, O.; Yitzchak, Y.; Bitton, R.; Nadler, A.; Shabtai, A. Incentivized delivery network of IoT software updates based on trustless proof-of-distribution. In Proceedings of the 2018 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), London, UK, 24–26 April 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 29–39.
6. Azaria, A.; Ekblaw, A.; Vieira, T.; Lippman, A. Medrec: Using blockchain for medical data access and permission management. In Proceedings of the 2016 2nd international conference on open and big data (OBD), Vienna, Austria, 22–24 August 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 25–30.
7. Zhang, Q.; Li, Y.; Wang, R.; Liu, L.; Tan, Y.; Hu, J. Data Security Sharing Model Based on Privacy Protection for Blockchain-enabled Industrial Internet of Things. *Int. J. Intell. Syst.* **2021**, *36*, 94–111. [CrossRef]
8. Xu, P.; Lee, J.; Barth, J.R.; Richey, R.G. Blockchain as Supply Chain Technology: Considering Transparency and Security. *Int. J. Phys. Distrib. Logist. Manag.* **2021**, *51*, 305–324. [CrossRef]
9. Wang, L.; Wang, Y. Supply Chain Financial Service Management System Based on Block Chain IoT Data Sharing and Edge Computing. *Alex. Eng. J.* **2022**, *61*, 147–158. [CrossRef]
10. Li, Q.; Ji, H.; Huang, Y. The Information Leakage Strategies of the Supply Chain under the Block Chain Technology Introduction. *Omega* **2022**, *110*, 102616. [CrossRef]
11. Kuo, T.-T.; Ohno-Machado, L. Modelchain: Decentralized Privacy-Preserving Healthcare Predictive Modeling Framework on Private Blockchain Networks. *arXiv* **2018**, arXiv:1802.01746.
12. Lakhani, K.R.; Iansiti, M. The Truth about Blockchain. *Harv. Bus. Rev.* **2017**, *95*, 119–127.
13. Zhao, Y.; Yu, Y.; Li, Y.; Han, G.; Du, X. Machine Learning Based Privacy-Preserving Fair Data Trading in Big Data Market. *Inf. Sci.* **2019**, *478*, 449–460. [CrossRef]
14. Xiang, Y.; Ren, W.; Li, T.; Zheng, X.; Zhu, T.; Choo, K.-K.R. A Multi-Type and Decentralized Data Transaction Scheme Based on Smart Contracts and Digital Watermarks. *J. Netw. Comput. Appl.* **2021**, *176*, 102953. [CrossRef]
15. Jing, N.; Liu, Q.; Sugumaran, V. A Blockchain-Based Code Copyright Management System. *Inf. Process. Manag.* **2021**, *58*, 102518. [CrossRef]

16. Luu, L.; Teutsch, J.; Kulkarni, R.; Saxena, P. Demystifying incentives in the consensus computer. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, 12–16 October 2015, Denver, CO, USA, 12–16 October 2015; pp. 706–719.
17. Xu, Y.; Shi, B. Copyright Protection Method of Big Data Based on Nash Equilibrium and Constraint Optimization. *Peer-Peer Netw. Appl.* **2021**, *14*, 1520–1530. [CrossRef]
18. Kumar, R.; Tripathi, R.; Marchang, N.; Srivastava, G.; Gadekallu, T.R.; Xiong, N.N. A Secured Distributed Detection System Based on IPFS and Blockchain for Industrial Image and Video Data Security. *J. Parallel Distrib. Comput.* **2021**, *152*, 128–143. [CrossRef]
19. Nasonov, D.; Visheratin, A.A.; Boukhanovsky, A. Blockchain-based transaction integrity in distributed big data marketplace. In Proceedings of the International Conference on Computational Science, Faro, Portugal, 12–14 June 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 569–577.
20. Zhou, J.; Tang, F.; Zhu, H.; Nan, N.; Zhou, Z. Distributed data vending on blockchain. In Proceedings of the 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Halifax, NS, Canada, 30 July–3 August 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1100–1107.
21. Strehle, E.; Maurer, M. The DibiChain protocol: Privacy-preserving discovery and exchange of supply chain information. In Proceedings of the International Conference on Model and Data Engineering, Tallinn, Estonia, 21–23 June 2021; Springer: Berlin/Heidelberg, Germany, 2021; pp. 231–247.
22. Nawaz, A.; Peña Queralta, J.; Guan, J.; Awais, M.; Gia, T.N.; Bashir, A.K.; Kan, H.; Westerlund, T. Edge Computing to Secure Iot Data Ownership and Trade with the Ethereum Blockchain. *Sensors* **2020**, *20*, 3965. [CrossRef]
23. Vincent, P.; Larochelle, H.; Bengio, Y.; Manzagol, P.-A. Extracting and composing robust features with denoising autoencoders. In Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, 5–9 July 2008; pp. 1096–1103.
24. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems, Proceedings of the 28th Annual Conference on Neural Information Processing Systems 2014 [(NIPS)], Montreal, QC, Canada, 8–13 December 2014*; Curran: Red Hook, NY, USA, 2015; Volume 27.
25. Salah, R.; Vincent, P.; Muller, X. Contractive auto-encoders: Explicit invariance during feature extraction. In Proceedings of the 28th International Conference on Machine Learning, Washington, DC, USA, 28 June–2 July 2011; pp. 833–840.
26. Bellet, A.; Habrard, A.; Sebban, M. A Survey on Metric Learning for Feature Vectors and Structured Data. *arXiv* **2013**, arXiv:1306.6709.
27. Luo, Y.; Liu, T.; Tao, D.; Xu, C. Decomposition-Based Transfer Distance Metric Learning for Image Classification. *IEEE Trans. Image Process.* **2014**, *23*, 3789–3801. [CrossRef] [PubMed]
28. Parameswaran, S.; Weinberger, K.Q. Large margin multi-task metric learning. In *Advances in Neural Information Processing Systems 23 (NIPS 2010)*; Lafferty, J., Williams, C., Shawe-Taylor, J., Zemel, R., Culotta, A., Eds.; NIPS: San Diego, CA, USA, 2010; pp. 1867–1875.
29. Szabo, N. Smart Contracts. 1994. Available online: https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html (accessed on 1 September 2022).
30. Wood, G. Ethereum: A Secure Decentralised Generalised Transaction Ledger. *Ethereum Proj. Yellow Pap.* **2014**, *151*, 1–32.
31. Das, S.; Ribeiro, V.J.; Anand, A. YODA: Enabling Computationally Intensive Contracts on Blockchains with Byzantine and Selfish Nodes. *arXiv* **2018**, arXiv:1811.03265.