Privacy and Integrity Protection for IoT Multimodal Data using Machine Learning and Blockchain

QINGZHI LIU, Wageningen University & Research, The Netherlands
YUCHEN HUANG, Wageningen University & Research, The Netherlands
CHENGLU JIN, Centrum Wiskunde and Informatica, The Netherlands
XIAOHAN ZHOU, Wageningen University & Research, The Netherlands
YING MAO, Fordham University, United States
CAGATAY CATAL, Qatar University, Qatar
LONG CHENG*, North China Electric Power University, China

With the wide application of Internet of Things (IoT) technology, large volumes of multimodal data are collected and analyzed for various diagnoses, analyses, and predictions to help decision-making and management. However, the research on protecting data integrity and privacy is quite limited, while the lack of proper protection for sensitive data may have significant impacts on the benefits and gains of data owners. In this research, we propose a protection solution for data integrity and privacy. Specifically, our system protects data integrity through distributed systems and blockchain technology. Meanwhile, our system guarantees data privacy using differential privacy and Machine Learning (ML) techniques. Our system aims to maintain the usability of the data for further data analytical tasks of data users, while encrypting the data according to the requirements of data owners. We implement our solution with smart contracts, distributed file systems, and ML models. The experimental results show that our proposed solution can effectively encrypt source IoT data according to the requirements of data users while data integrity can be protected under the blockchain.

Additional Key Words and Phrases: blockchain, machine learning, Internet of Things, multimodal data, privacy, integrity.

ACM Reference Format:
Qingzhi Liu, Yuchen Huang, Chenglu Jin, Xiaohan Zhou, Ying Mao, Cagatay Catal, and Long Cheng. 2023. Privacy and Integrity Protection for IoT Multimodal Data using Machine Learning and Blockchain. 1, 1 (December 2023), 19 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

With the wide application of Internet of Things (IoT) technology, such as smart buildings, precision agriculture, and smart warehouses, large volumes of multimodal data are generated and shared among different users [1] [2]. There are numerous multimodal data collected from IoT systems, such as air temperature sensors, microphones, and surveillance

*Corresponding author: Long Cheng, North China Electric Power University, China. (lcheng@ncepu.edu.cn)

Authors' addresses: Qingzhi Liu, Wageningen University & Research, The Netherlands, qingzhi.liu@wur.nl; Yuchen Huang, Wageningen University & Research, The Netherlands, yuchen.huang@wur.nl; Chenglu Jin, Centrum Wiskunde and Informatica, The Netherlands, chenglu.jin@cwi.nl; Xiaohan Zhou, Wageningen University & Research, The Netherlands; Ying Mao, Fordham University, United States, ymao41@fordham.edu; Cagatay Catal, Qatar University, Qatar, ccatal@qu.edu.qa; Long Cheng, North China Electric Power University, China, lcheng@ncepu.edu.cn.

cameras [3]. These data can be collected and analyzed for various diagnoses, analyses, and predictions to help decision-making and management [4]. Especially in Machine Learning (ML) technology, data plays a vital role in the performance of the ML model. However, the privacy and integrity of the IoT multimodal data are becoming a challenge for IoT-based applications. Specifically, despite all the benefits of IoT systems, data owners are reluctant to share their data with other data users, due to privacy issues such as lack of trust and data leakage. At the same time, data users are hesitant to use the shared data, because the integrity of multimodal data can hardly be guaranteed, which could negatively affect the accuracy and safety of data analysis. Therefore, in front of these challenges, it is urgent to build a data management system that can effectively cope with both data privacy and integrity issues. Although there is much research on data security, the research on privacy and integrity of multimodal data is relatively limited. In this paper, we propose a system for protecting the privacy and integrity of IoT multimodal data using blockchain and ML.

The security challenges that we aim to solve include the integrity and privacy of IoT multimodal data. Data integrity refers to the assurance of data accuracy, completeness, and consistency [5]. Once data integrity is protected, the source data should remain accurate and reliable during data storage or data access. Tampering with data integrity is a highly damaging cyber-security issue. Data tampering can occur unintentionally or maliciously by removing specific data entries or altering particular sections of data. Once data integrity is broken, restoring the original data would be difficult [6]. Therefore, using source data without understanding its integrity condition can put data-driven sectors at risk. Another aspect of our research is data privacy. Data privacy is a branch of data protection for the proper handling of sensitive data, such as personal data or confidential data. Leaking private data can cause significant loss to the data owner. According to the research [7], due to a lack of trust in how data is used, farm owners become hesitant to adopt IoT technologies to their farms. Therefore, there needs a privacy protection mechanism that can make data owners trust the data users to process and analyze their data.

Some existing solutions use blockchain for data integrity protection. The system of [8] adapts blockchain and decentralized technology to improve data sharing for healthcare applications. The work of [9] proposes a solution using blockchain technology to share health data safely. However, most of these solutions do not consider the following two key points.

- How to protect the privacy and integrity of multimodal data at the same time?
- How to make data owners be able to control the protection level of data privacy and integrity?

To resolve the issues above, we leverage blockchain and ML to protect multimodal IoT data in a decentralized system. On the one hand, as for data privacy-related issues, we use ML to analyze the content of data and adopt Differential Privacy (DP) mechanism [10] to encode source data according to the protection requirement of data owners. On the other hand, we store data on a distributed system, and the related key values are saved on smart contracts to guarantee data integrity. At the same time, to protect the hidden information of multimodal data, we use ML to analyze the users' requirements and make corresponding changes to the smart contract. In general, the main contributions of our work are summarized as follows.

- We propose a solution for protecting the integrity and privacy of IoT multimodal data. Our system protects data integrity through distributed systems and blockchain technology and protects data privacy using DP.
- We leverage ML techniques to control the privacy protection level for data owners. To the best of our knowledge, this is the first system that integrates blockchain technology and ML to control the protection level for data integrity and privacy at the same time.

- Our solution is extensively evaluated based on the data that is generated from ML models and a greenhouse simulator. The experimental results show that the proposed solution can achieve the expected performance for privacy and integrity protection.

The remainder of this paper is organized as follows. The related work is discussed in Section 2. The detailed system design and implementation are presented in Section 3. The experimental results are discussed in Section 4. We discuss the future work in Section 5. Finally, we conclude the paper in Section 6.

2 RELATED WORK

2.1 Blockchain

2.1.1 Blockchain Architecture. Blockchain is a distributed data storage system that records all transaction details in the peer-to-peer network [11] [12]. All the participants in the network can hold a copy of the data. Therefore, no central authority nor a single node can control the entire network, thus effectively achieving decentralized data protection.

In blockchain, all the information related to transaction records is stored and tracked through a sequence of “blocks" effectively forming a chain. Each block consists of information divided into a block head and block body. The block head contains information such as the current block hash, parent block hash, and timestamp. The block body contains transaction records between the community users. The block hash is a string created by a cryptographically secure hash algorithm, which is used for maintaining the immutability of data. Therefore, blockchain offers integrity protection, as data tampering causes alteration of the block hash leading to mismatches in the blocks.

2.1.2 Smart Contracts. Currently, numerous cryptocurrencies use blockchain technology [13]. A cryptocurrency called Ethereum [14] offers “smart contract", which can perform distributed computing on blockchain. Smart contracts on Ethereum are a set of executable programs that are stored on a blockchain-based platform [15]. A smart contract operates as a contract by executing and enforcing agreements between multiple parties. For example, transferring funds from one party to another party when a set of conditions is met. Smart contracts are stored on a blockchain, which has the benefits of security and immutability protection that blockchain offers.

2.1.3 Interplanetary File System. Interplanetary file system (IPFS) [16] is a peer-to-peer storage dedicated network where data is stored and shared in a distributed file system. When data is stored in IPFS, the data content is split into smaller chunks and hashed to give the content a unique content identifier (CID). IPFS system is composed of three fundamentals. Namely, content identification via content addressing, content linking using directed acyclic graphs (DAGs), and content discovery using distributed hash tables (DHTs). IPFS uses CID to access its contents instead of using location-based addressing, such as a URL link. CID is created by hashing the data content, and a hash value is unique to its data content. Therefore, different content produces a different unique CID. DHTs are used to find peers hosting the content to access the data content. A hash table is a database of keys with their corresponding values. By distributing the hash table throughout the network, users can find their peers hosting the content by providing the CID.

2.2 Differential Privacy

IoT multimodal data has much meaningful information for various analyses. To hide the private information of source data, methods such as differential privacy (DP) have been developed [17] [18]. After encrypting data by DP, data can be shared with other data users without jeopardizing the privacy of the data owner.

To further explain the DP concept, assume two identical numeric databases D1 and D2. A randomized mechanism M gives ϵ-DP if for all databases such as D_1 and D_2 differing on at most one row and all $S \subseteq Range(M)$. Suppose $M(D_1)$ is the mechanism's output M after applying the query function to the database D_1. Pr is the probability distribution of $M(D_1)$. The probability distribution between the randomized mechanism output of the two databases is similar, making it impossible to identify which of the two data sets the query was executed. The differential privacy algorithm must fulfill the requirement of being able to produce outputs with similar probability distribution as:

$$Pr[M(D_1) \in S] \leq exp(\epsilon) \times Pr[M(D_2) \in S] \tag{1}$$

There are several mechanisms to implement DP. Laplace mechanism [19] is the most popular mechanism [20] and will be used in the system to provide privacy protections. The Laplace mechanism adds randomized noise to the query output $f(D)$ drawn from the Laplace distribution. The standard deviation of the generated Laplace distribution is based on the privacy budget (ϵ) and sensitivity (Δs). The mechanism itself can be described as:

$$M(D) = f(D) + Lap(\frac{\Delta s}{\epsilon}) \tag{2}$$

, where the sensitivity Δf is calculated from Equation 3. The privacy budget ϵ is used to manage the privacy level during data privatization.

The sensitivity measures the maximum change in one data set with at most one element, where the possible changes are bound by the local data set. The sensitivity when function f is operating on one data set D_1 can be described as:

$$\Delta f_S = \max_{D_2} \|f(D_1) - f(D_2)\|_1 \tag{3}$$

, where D_1 is the data set and D_2 is another data set with at most one different element relative to D_1.

A DP mechanism can be used in two different adversarial settings: global and local. The global DP mechanism assumes that the data curator is trusted and all the original data can be given to the curator. Any query from a (malicious) data analyst has to go through the data curator, so the data curator can inject noise in the processed data query to privatize the original data. Local DP mechanism works with a stronger adversarial model, i.e., there is no trusted data curator. Hence, the noise injection in a DP mechanism has to be done by individual data providers. In our work, we will use the local DP mechanisms because, in our system, there is no trusted data curator that can serve queries on the data.

2.3 Solutions for Privacy Protection

Many blockchain-related solutions are proposed for security and privacy protection [21] [22] in various applications, such as healthcare and smart grid. The paper [23] proposes a blockchain-based access control framework for cloud service to protect security and privacy. The solution uses matrix encryption to protect the privacy of users' activities and uses fully homomorphic encryption to protect the privacy of data content. Meanwhile, the solution leverages a smart contract for access control and another smart contract for audit users' activities. Although this solution can verify and audit the users' activities and protect their privacy, it lacks the flexibility to manage the privacy protection level for the users. To cope with the problem that blockchain cannot preserve data privacy due to its decentralized manner, the system proposed by [8] uses data encryption. By adopting this solution, the raw data files with the associated ID could be encrypted before uploading to IPFS and smart contract. The solution can efficiently prevent any participant nodes from inspecting the block and obtaining sensitive private data. However, this means that a separate system needs to be designed for the data encryption, leading to an off-chain solution that is less desirable due to the involvement of trusted third parties. A solution for privacy-preserving data sharing is proposed by [24]. The system uses cloud services to store

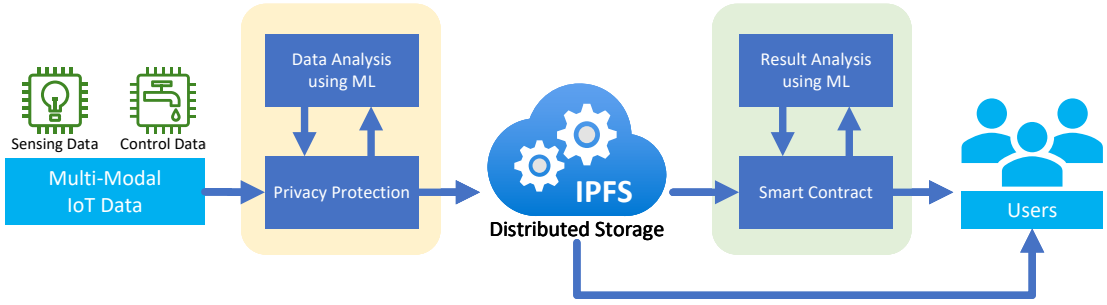


Fig. 1. An overview of the system.

sensitive data while using a tamper-proof consortium blockchain to manage access control, including access permissions, actions (read, write, or copy), and duration to authenticated sharing. However, this solution would lead to centralization by using cloud storage or a private server. The paper [25] proposes a blockchain-based access control framework with privacy protection. The main method of the framework is encrypting and storing the data permission in blockchain and building an access control permission of data in a cloud environment. The implementation of the framework builds access control, authorization, and authorization revocation. This paper concludes that the authorization and access control performance can be configured by blockchain. However, the solution cannot differentiate the privacy protection level of different users. The paper [26] proposes a blockchain-based privacy protection and security sharing scheme. In this scheme, data is divided into data owners, data requesters, and authorization institutes. The authorization institute grants a hash anonymous ID and access rights, which are used to verify the access rights of the data requester. Meanwhile, the system checks whether the data meets the requirements without disclosing data privacy. However, this scheme is only suitable for the data that can be stored in centralized or authorized institutes, which cannot guarantee data integrity.

3 SYSTEM DESIGN AND IMPLEMENTATION

In the system, we assume the data is multimodal, including sensing data, camera data, and control data of actuators. The data owners want to share the data with others while needing to protect the sensitive information in the data. The data users will leverage the shared data for analysis while needing a guarantee of the data integrity. For example, the IoT systems of precision agriculture produce much multimodal data. On the one hand, farm owners have source data that are collected from sensors and actuators about some growth and yield of crops on farms. On the other hand, data users, such as companies and universities, are interested in researching the data to improve agricultural operations and yields. In this case, some statistical information of the data should be protected according to the requirements of data owners, because leaking such data could potentially harm the benefit of farm owners. Meanwhile, the shared data should have sufficient integrity compared to source data, so that the information of data can be accurately analyzed by data users. Therefore, our system aims to protect privacy and integrity with the following requirements.

- The information of data should be integrity enough for data users to analyze and use.
- The data, including the hidden information of multimodal data, should be protected according to a quantified level controlled by the data owners.

To fulfill these requirements, our system uses a blockchain to protect data integrity and ML to protect data privacy. The system mainly has three components, including encoding data, data storage, and managing smart contracts. The workflow of the system is as follows and is shown in Figure 1.

(1) The data is analyzed by ML and encoded by DP. (Section 3.1)
(2) The encoded data is stored on IPFS and produces a CID for each data item, and the CID is saved on the Ethereum blockchain through the smart contract. (Section 3.2)
(3) The system manages smart contracts to fulfill the requirements of data owners and users. (Section 3.3)

3.1 Encoding Data

3.1.1 Differential Privacy. After receiving the data from IoT sensors, our system encodes the data using a DP mechanism. In our implementation, we use the Laplace algorithm as the DP approach [27]. Specifically, the Laplace mechanism takes random numeric values from the Laplace distribution and produces output with a similar distribution. The noise is injected into each entry of the data set. For different privacy protection level, the amount of noise can be tuned by ϵ value. A low ϵ value can provide high privacy protection by injecting a high amount of noise, while a high ϵ value can provide low privacy protection.

DP is not the only method that can protect data privacy. However, compared with other traditional secure computation methods, like garbled circuits, and secure multi-party computation, DP allows an efficient implementation on resource-constrained devices and still provides a well-defined privacy guarantee. Meanwhile, DP is generally lightweight and its impact on data aggregation accuracy can be well understood by users. In DP techniques, we chose to use the Laplace algorithm because it is the most widely used DP scheme and it fits the hypothetical scenario where we want to protect the privacy of individual data sources while preserving the utility of the statistical information of all the data.

Many evaluation metrics corresponding to various privacy-enhancing techniques have been proposed as a way to define or quantify privacy, such as the confidentiality used in secure computation [28], differential identifiability [29], membership privacy [30], k-anonymity [31]. There are no closed-form formulas to translate the metric we use to other privacy metrics, since they are defined in different frameworks. In our evaluation, we use ML accuracy as one way to demonstrate the protection of DP, which is a clear level for users that shows how well our system protects meaningful information of source data being derived by attackers.

3.1.2 Controlling Protection Level. Although the Laplace mechanism tunes noise level, the protection level for data owners cannot be controlled. Therefore, our system offers the option for the users to control the privacy protection level as follows.

- **Step 1**: Data owners select the expected results they want the data can be generated in data analysis. For example, the privacy protection level can be defined as the prediction accuracy of a data set.
- **Step 2**: The data is analyzed by various ML models. For example, using historical data to predict future data.
- **Step 3**: The system compares the ML-based analyzing results to the selected privacy protection level. If the accuracy of the analysis result is higher than the required protection level, then the system increases the noise level by decreasing ϵ as explained in Section 2.2.
- **Step 4**: The system executes from Step 2 again, and continues these steps until the analyzing accuracy is lower than the requirements of the data owners.

As a case study, we generate a series of IoT sensing data from a smart greenhouse simulation. The aim is to prevent data users from estimating some sensing data based on the other sensing data. For example, there are several types

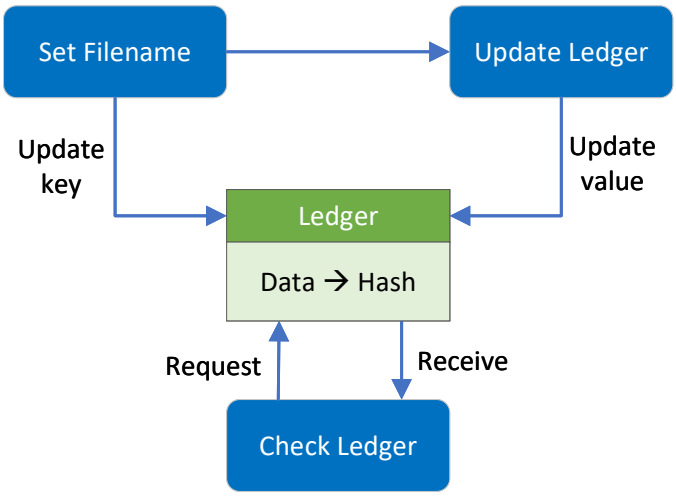


Fig. 2. Workflow of smart contract.

of sensing data in a smart greenhouse. The data owner would like to share only some of the sensing data and hide some others. To cope with this problem, we leverage a Deep Neural Network (DNN) model to explore the relation between different sensing data [32]. The model is first trained by the historical sensing data. In the training data set, the input data types to the DNN model are the data types that the data owners agree to share, and the output data types of the DNN model are the data types that the data owners will hide. In the experiment, we set a required accuracy of prediction. As long as the accuracy of the DNN model is higher than the required accuracy of the user, we keep decreasing ϵ.

In our system, the data owner has all the data. Therefore, it can guarantee the data owner has more data for training the DNN model. Suppose the data owner and data user have the same approach for building DNN models. More accurate data can guarantee better performance of the DNN model. In addition, the data obtained by data users is encrypted by DP. This means the data owned by the data owner has a higher quantity and better accuracy than the data of data users. Therefore, the performance of the DNN model of the data owner is better than data users, and we believe the DNN trained by the data user cannot detect the hidden information.

3.2 Data Storage

3.2.1 IPFS. To protect data integrity, we store data in blockchain. To reduce the size of data saved in the blockchain, we store source data in IPFS and make hash code for the data. The CID value generated by hashing is then saved on the smart contract, which is deployed on the blockchain. The hashing CID allows the system to verify data integrity without saving raw data on the Ethereum blockchain. In our implementation, the sha256 algorithm is chosen for data hashing. In addition, the Merkle tree data structure is implemented for verification. In this way, the whole operation is decentralized and thus provides integrity protection for the data.

3.2.2 Smart Contract. After saving data to IPFS, the CID is saved in a smart contract implemented by the Ethereum blockchain. When the data is stored in IPFS, the data is split into small chunks and hashed to give the data a CID. By providing CID, users on the Ethereum blockchain can obtain access to the data. By saving the CID using the smart

contract, the data integrity is protected while maintaining the whole process decentralized. Data users must fulfill the smart contract signed with the data owner for obtaining the CID before accessing the data file on IPFS. Meanwhile, data users can verify the data integrity by inspecting the hash value on the smart contract. The data on the Ethereum blockchain and IPFS is practically immutable, thus protecting data integrity.

The workflow of the smart contract is shown in Figure 2. The contract contains an empty data type mapping called ledger, which acts as a hash table consisting of a key and a corresponding value. The ledger is created to save CID with its corresponding identification, such as filenames. The interaction between the ledger and users mainly includes two functions, namely `set_filename` and `updateLedger`. Firstly, a data identification string is set by using the function `set_filename` creating a key with empty values in the ledger. Secondly, the hash string is passed through the function `updateLedger` with its corresponding identification to save or update the hash value. In our implementation, to protect data privacy and integrity, both functions in the smart contract could only be used by the data owner. Meanwhile, to access the CID, data users can call a function called `check_ledger` to view the data by inputting their corresponding identification.

3.3 Managing Smart Contract

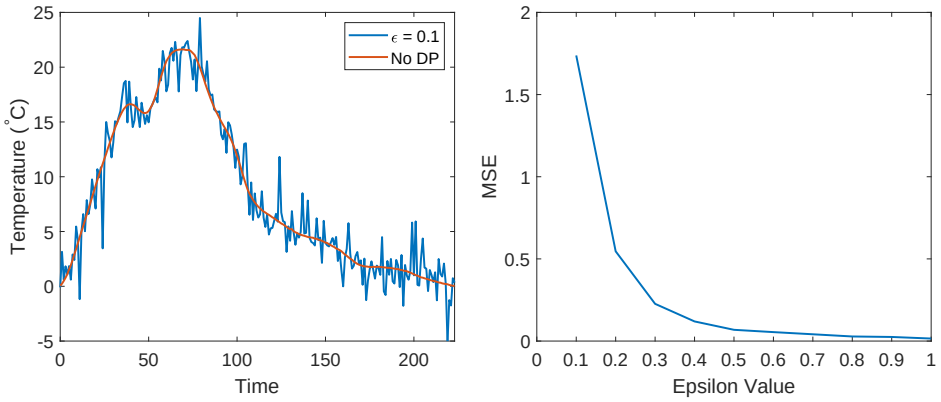
For a series of control data, we need not only to protect the data itself, but also protect the privacy information that these control data can generate. For example, in a smart greenhouse, a batch of control data can make a greenhouse produce an amount of yield. This yield information is confidential to some growers. Therefore, although the control data can be shared, the related yield that is correlated to the control data must be protected.

To control the privacy protection level of control data, we leverage Deep Reinforcement Learning (DRL) model to explore the optimal control results [33][34]. Then, according to the optimal control results, the system selects which data to share with users. The detailed working steps are as follows.

- **Step 1**: Data owners define the expected control results that can be visible to data users after data analysis.
- **Step 2**: The system inputs the control data and related sensing data to the DRL model, and calculates the optimal control results.
- **Step 3**: The system compares the expected results of data owners and the optimal results. If the expected result is lower, the system randomly removes some samples of control data that are shared by the smart contract.
- **Step 4**: The system executes from Step 2 again, and continues these steps until the optimal control result is lower than the expected value of the data owner.

The control data can also be protected by the same approach as in Section 3.1. Meanwhile, we find the control results are different using different parts of control data. The data selection only happens when making a contract between the data owner and the user. It is unknown which part of the control data is selected before making the smart contract. Therefore, by using DP in sensing and controlling data beforehand, we cannot completely manage whether the selected data can generate optimal results. In addition, the source data has already been saved in the blockchain before making the smart contract. To protect the data integrity, it is not allowed to further change the data after uploading to the blockchain. Therefore, we leverage the DRL model to explore the optimal control result, and dynamically select which controlling data can be shared.

In our implementation, we use Deep Q-Network (DQN) [35] as a DRL model. DQN model is designed by combining a classic Q-learning reinforcement learning algorithm with DNN. It is widely used to solve decision-making problems. Based on the architecture design of our system, all types of DRL models, such as Asynchronous Advantage Actor-Critic



(a) Example encoding using Laplace algorithm with $\epsilon = 0.1$. (b) MSE of sensing data with ϵ from 0.1 to 1.0.

Fig. 3. Use the ϵ value of the Laplace algorithm to control the noise level inserted into source data.

(A3C) [36] and Proximal Policy Optimization (PPO) [37], can be used for managing smart contracts. In this work, we use DQN to validate the correctness of the system, because it is a widely used DRL solution for various applications. For our testing scenario, the optimal result of the DQN model is enough for system validation. The other performance of the DRL model, such as convergence speed, is not needed for the system implementation and testing.

4 EXPERIMENTS

The system experiments are composed of evaluations of privacy and integrity protection. The data privacy tests aim to evaluate the DP encoding algorithm and ML-based data analysis. The data integrity tests aim to evaluate whether the smart contract operates correctly in applications.

4.1 Evaluation on Differential Privacy

This DP experiment is to evaluate the protection for the data sets rather than the statistics information of data sets. This means that the statistical information, such as the mean and variations of data sets, are visible to preserve the utility of the data for further analysis in this experiment.

4.1.1 Testing Data. We generate two types of data for evaluating the performance of DP, including data from linear regression and clustering models.

To generate data of linear regression, we make feature and target variable values based on a linear regression model using the "make_regression" function from "scikit-learn" library, in which the linear regression model can predict the "Target" values based on the "Feature" values. The noise and features of the model are 10 and 1 respectively. The size of the generated data is 1000 samples.

To generate data of clusters, we make two isotropic Gaussian blobs for clustering using the "make_blobs" function from "scikit-learn" library, in which each blob position has two values "Position X" and "Position Y". The centers, features, and cluster standard deviation are [(0,0), (5,5)], 2, and 1.5 respectively. The size of the generated data is 1000 samples.

Table 1. Error-metric of the query means for all data sets.

	ϵ = 0.1		ϵ = 5		ϵ = 50	
Data Set	Mean	MAE	Mean	MAE	Mean	MAE
Feature	349	38.9	320	9.9	309.6	0.5
Target	65.4	5.8	61.2	1.7	59.7	0.1
Position X	3.7	1.1	2.9	0.3	2.5	$2.3 * 10^{-3}$
Position Y	4.2	1.6	2.8	0.2	2.6	$6.9 * 10^{-3}$

Table 2. Error-metric of the query median for all data sets.

	ϵ = 0.1		ϵ = 5		ϵ = 50	
Data Set	Median	MAE	Median	MAE	Median	MAE
Feature	335.1	26.3	312.2	3.3	307.6	1.2
Target	44.7	5.8	14.5	0.4	59.3	$6.4 * 10^{-3}$
Position X	7.9	5.3	2.7	0.1	2.6	$8.4 * 10^{-3}$
Position Y	$3.0 * 10^{-2}$	2.6	2.6	$2.2 * 10^{-2}$	2.4	0.2

Table 3. Error-metric of the query variance for all data sets.

	ϵ = 0.1		ϵ = 5		ϵ = 50	
Data Set	Variance	MAE	Variance	MAE	Variance	MAE
Feature	125540.5	115124.1	29372.0	18955.5	10794.9	378.4
Target	2811.0	2613.0	657.5	459.5	204.6	6.6
Position X	51.7	43.1	16.4	7.8	8.6	$8.9 * 10^{-2}$
Position Y	51.6	42.8	16.4	7.7	8.9	0.2

4.1.2 Testing Results. To measure the efficacy of added noise to source data, we use Mean Absolute Error (MAE) $\frac{1}{n}\sum_{i=1}^{n}|X_i - Z_i|$ and Mean Squared Error (MSE) $\frac{1}{n}\sum_{i=1}^{n}(X_i - Z_i)^2$, in which X is the original value of source data, Z is the value with LP noise, i is the order number of the sample data, and n is the total number of data.

In the experiments, we use the Laplace mechanism with different ϵ to the data set. An example result is shown in Figure 3. In Figure 3(a), we can see the source data is perturbed by the noise data of the Laplace mechanism. Meanwhile, we change the value of ϵ from 0.1 to 1.0. It can be seen from Figure 3(b), that the MSE decreases as the value of ϵ increases.

The testing results using data from regression and clustering models are shown in Table 1, 2, and 3. The tables show the MAE and three statistical values mean, median, and variance. According to the results, the data statistics perturbed with a low $\epsilon = 0.1$ have a higher error-metric value compared to data perturbed with a high $\epsilon = 50$. Meanwhile, the output of the data entries generated by the Laplace mechanism with $\epsilon = 5$ has a similar mean, median, and variance value compared to the source data. This means the Laplace mechanism fulfills the protection requirement of data owners while keeping the statistics information. Based on the experimental results, we conclude the amount of privacy protection level of data owners is adjustable by setting the ϵ value, while the data statistical values are kept well. The lower the ϵ, the greater the privacy provided, resulting in a reduction in data utility.

4.2 Evaluation on ML-based Protection Level

4.2.1 Testing Data. To evaluate whether the ML-based data analysis can be used to control privacy protection levels, we use two typical ML models that are widely used for data analysis, including the linear regression model [38] and Support Vector Machine (SVM) classification model [39]. The reason that we select the linear regression model and

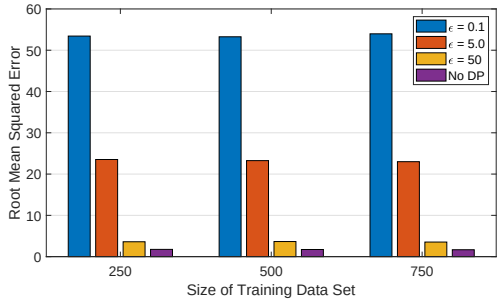


Fig. 4. Use regression model for controlling the privacy protection level.

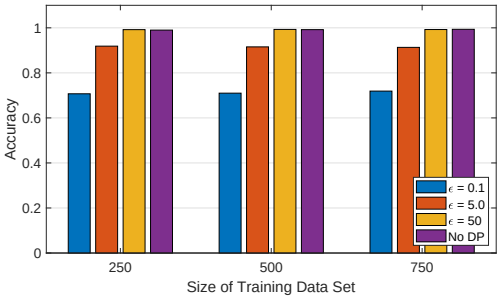


Fig. 5. Use classification model for controlling the privacy protection level.

SVM model for evaluation is that these ML models are widely used for prediction and classification. Meanwhile, these models are relatively easy to implement with high accuracy, which is very suitable as a case study experiment for our system. The application of other more complicated ML models is out of the scope of this work. The testing data for the linear regression model and the SVM model is the same as the DP evaluation testing which is explained in Section 4.1.1.

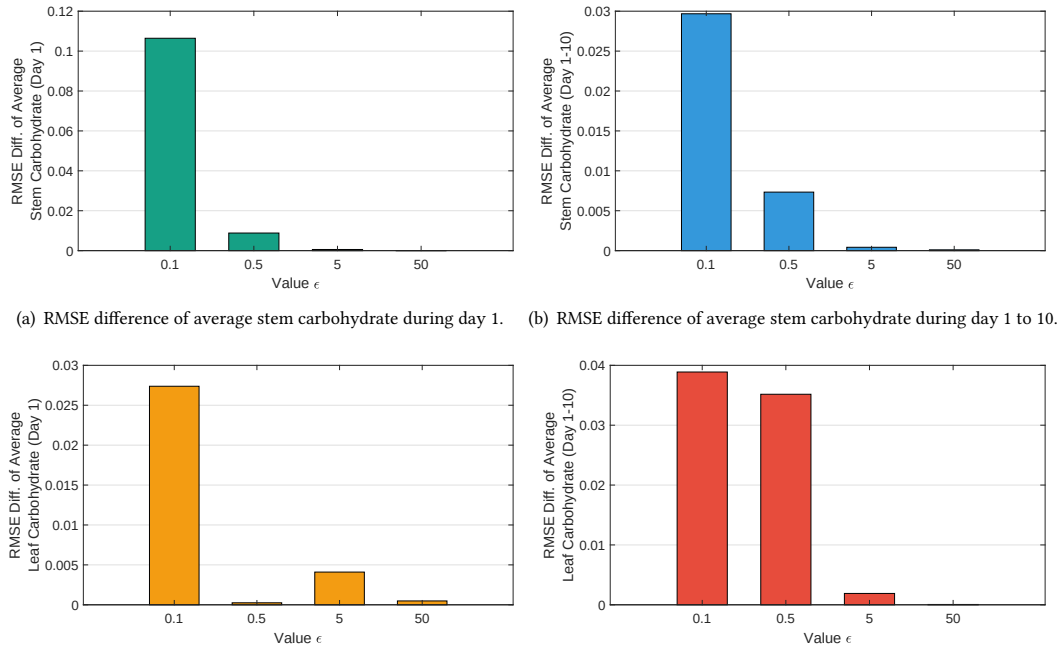
To evaluate whether DNN models can also be leveraged to control privacy protection levels, we generate data from GreenLight simulator [40], which simulates data for a greenhouse environment with tomatoes. The data includes indoor climate states, crop states, outdoor climate states, and control values. A DNN model is built to predict the crop features based on the other greenhouse states. By GreenLight simulator, we generate 10 sets of data. Each data set has 4320 entries, which represents continuous data collection every 1 hour for 180 days in a greenhouse. We use 9 sets of data for training the DNN model, and 1 set of data for testing.

4.2.2 Testing Results. We train the linear regression and SVM models using the data with various DP levels. Both ML models are trained using three different ϵ values, including 0.1, 5.0, and 50. As explained in Section 4.1, $\epsilon = 0.1$ represents the highest privacy protection and 50 represents the lowest. For each ϵ value, different training samples were used for model training, namely 250, 500, and 750. The performance of the linear regression model is measured by the root of MSE (RMSE), and the performance of the classification model is measured by accuracy.

The results of the regression model are shown in Figure 4. The model with the lowest ϵ value 0.1 has the highest RMSE, while the highest ϵ value 50 makes the lowest RMSE. A different training sample size was also used to train the

Table 4. The parameters of DNN model.

Model	Parameter	Value
DNN	Neurons in Input Layer	203
	Neurons in Output Layer	276
	Neurons in Hidden Layer	406
	Number of Hidden Layer	4
	Activation Function	ReLU
	Loss Function	MSE
	Learning Rate	0.0001
	Epochs	1000



(a) RMSE difference of average stem carbohydrate during day 1.

(b) RMSE difference of average stem carbohydrate during day 1 to 10.

(c) RMSE difference of average leaf carbohydrate during day 1.

(d) RMSE difference of average leaf carbohydrate during day 1 to 10.

Fig. 6. Use DNN model for controlling the privacy protection level on the stem and leaf carbohydrate values.

model for each epsilon setting. With different sizes of the training data sets, the RMSE trends affected by ϵ value are the same.

The results of the SVM model are shown in Figure 5. The model with the lowest ϵ value 0.1 has the lowest accuracy, while the highest ϵ value 50 has the highest accuracy. A different training sample size was also used to train the model for each epsilon setting. With varying sizes of the training data sets, the accuracy trends affected by the ϵ value are the same. The performance is heavily reduced after a high noise injection. As the privacy protection level decreases, the model is able to produce similar performances to the model trained with original data.

Table 5. Detail content of a deployed smart contract.

Deployment Action	Details
Transaction Hash	0xfc12a90838b90ea8be57230dc3f7f1ff0c686a2710e48bd7533c67bf2233086e
Block Number	10537518
Timestamp	Apr-20-2022 % 01:50:24 PM +UTC
Contract Address	0x62334b844e3d352d423d1136dffe690d8690b311
Transaction Cost	409680

After testing the system using the above two typical ML models, we use a DNN to evaluate whether the requirement of the data owner can be fulfilled by controlling the ϵ value. The input layer of the DNN model has 203 state values from GreenLight simulation. Specifically, the input data includes 20 initial indoor climate states, 3 initial crop states, 7 outdoor climate states $\times$ 12, and 8 control values $\times$ 12, in which value 12 represents the number of samples per hour. The output layer of the DNN model describes the prediction values of the greenhouse environment and crop growth in GreenLight simulation. Specifically, the output data includes 20 indoor climate states $\times$ 12, and 3 crop states $\times$ 12, in which value 12 represents the number of samples per hour. The other key parameters of the DNN model are shown in Table 4.

The testing results are shown in Figure 6, which clearly show an increase of RMSE as the value of ϵ decreases. This trend on different testing datasets is the same, including the average stem carbohydrate of day 1 in Figure 6(a), the average leaf carbohydrate of day 1 in Figure 6(c), the average stem carbohydrate of day 1 to 10 in Figure 6(b), and the average leaf carbohydrate of day 1 to 10 in Figure 6(d). In addition, this trend is the same as the results of the other ML models in Figure 4 and Figure 5. In Figure 6, some data points show a different trend, such as the data points $\epsilon = 0.5$ and $\epsilon = 5$ in Figure 6(c). This is mainly due to the limited training data, which cannot train a DNN model to predict with very high accuracy. To cope with this issue, the system can use more data for training the DNN model, and calculate the average value of the RMSE difference in several ϵ to smooth and generate a clear trend.

The results of the above ML and DL models show that, on the one hand, a high level of privacy protection (low ϵ setting) will negatively impact the model's performance, which reduces data utility. On the other hand, low privacy protection (high ϵ setting) will less affect the performance of the ML models. Therefore, we can conclude, by leveraging ML models, the privacy protection level can be controlled by changing the ϵ value step by step. In this way, we could fulfill the request of data owners about how much privacy level they want to apply to the source data.

4.3 Evaluation on Data Storage

Blockchain and smart contracts play a crucial role in data integrity protection for the current system design. We deploy and interact with the smart contracts on the Ethereum blockchain. In this experiment, we mainly test the functionality of smart contracts.

The contract is deployed on the Ethereum Rinkeby testnet to simulate the scenario of using Ethereum blockchain. The deployment resulted in a transaction hash which was used to find details about the contract detail information. The deployment details are shown in Table 5. After the deployment, a data file called `mockdata_linear` was uploaded to the IPFS, and then save the corresponding CID value on the smart contract. In the first upload interaction as shown in Table 6, the first transaction saves the filename on the contract, and the second transaction saves the CID value. In the second upload interaction as shown in Table 7, a new data file was uploaded to the IPFS.

Table 6. Details of the first upload interaction. The contract interaction details include CID hash and the related file name.

First Upload Interaction	Interaction Details
Transaction Hash 1	0xdc5814e1981253e62ed1a04e3f742c894ea734e647ee06c75e1834cf95ede27f
Filename	`mockdata_linear`
Transaction Cost	27062
Transaction Hash 2	0xd941b32e4b454cdd416f28150c96e5f346b360cd0d70ca50b156389be52216bb
CID hash	QmPqbL9S4sFbvcy5snxfptiyAGB5sDtHr8RrR1mS3yqGKk
Transaction Cost	94177

Table 7. Details of the second upload interaction. The contract shows the unique CID value after uploading a new file.

Second Upload Interaction	Interaction Details
Filename	`mockdata_linear`
CID hash	QmVQJXpkUNmQWUSFgqsiNXLsRP8xZxXmr9VBTZRPqXjkb3

The contract deployment details include the transaction hash, which is used to find the deployment details. The block number represents the identification of the blocks in the blockchain. The timestamp indicates the time when the transaction is made. The contract address indicates the public address at which the contract is called for transactions. Finally, transaction cost represents the price related to the contract.

By using blockchain, data integrity is protected once the file has been uploaded to IPFS. Changing data would result in a different CID hash which prevents the original data from being altered. As shown in the result, a data file has been successfully uploaded to IPFS while using a smart contract to save/access the unique CID. The whole process can be traced and verified through the transaction hash. A modified data file produced a different CID value despite being similar to the first original data file. In this way, data integrity protection is guaranteed by the IPFS and Ethereum blockchain.

The Ethereum-based smart contract is a case study for our system. Any other blockchain platforms can be used in the system. We selected an Ethereum-based smart contract because it has relatively good software API support and community, which is easy to implement.

4.4 Evaluation on Smart Contract

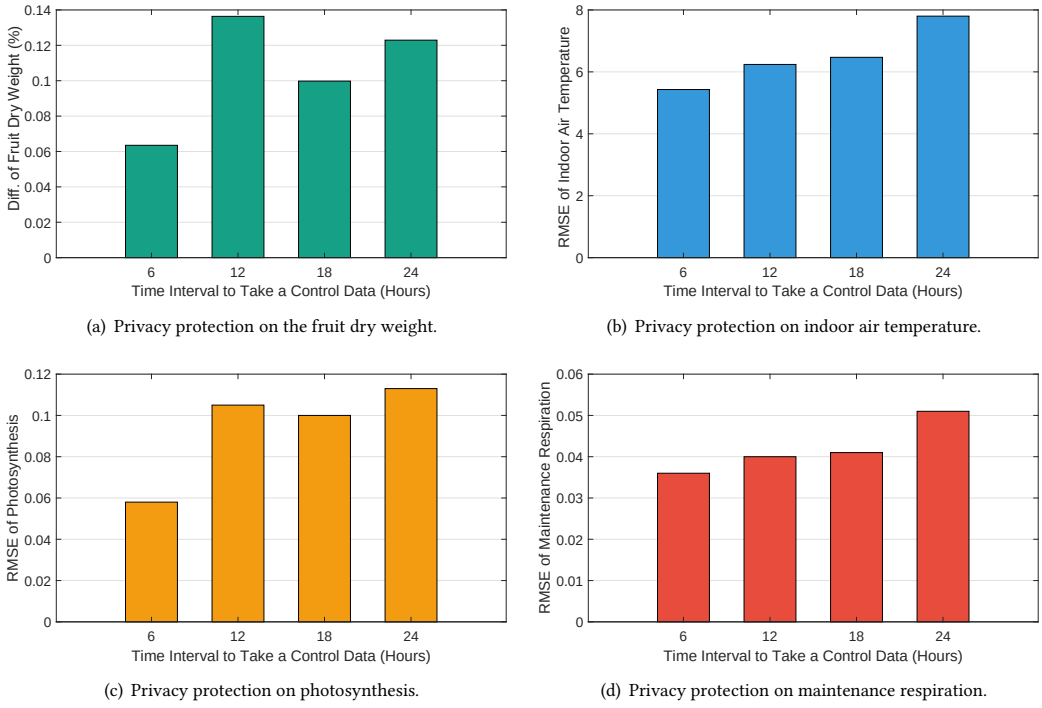
In the final experiment, we evaluate whether the DRL model can manage the smart contract. We aim to prevent the selected control data from producing crop growth that is close to optimal values.

4.4.1 Testing Data. We generate simulation data of a smart greenhouse using GreenLight. The simulation data has three categories of IoT data, including environmental sensing data, crop sensing data, and greenhouse control data. Specifically, the environmental sensing data include temperature, relative humidity, CO_2, light intensity, and so on. The crop sensing data include plant height, leaf size, crop size, and so on. The greenhouse control data include opening windows, turning on lights, setting ventilation systems, setting irrigation systems, and so on. We generate 48 entries of data, which represents continuous data collection every 1 hour for 2 days.

As explained in Section 3.3, the DQN model calculates the optimal control results based on the input control data and related sensing data. The DQN model is trained on the GreenLight simulator until the reward value is converged. The structure of the DQN model is as follows. The input data to the DQN model includes 18 greenhouse state values, including time step, roof ventilation, outdoor temperature, outdoor air humidity, outdoor carbon dioxide concentration,

Table 8. The parameters of DQN model.

Model	Parameter	Value
DQN	Neurons in DNN Input Layer	18
	Neurons in DNN Output Layer	7
	Neurons in DNN Hidden Layer	54
	Number of DNN Hidden Layer	3
	DNN Activation Function	ReLU
	DNN Loss Function	MSE
	Gamma	0.95
	Initial Epsilon	1.0
	Min Epsilon	0.02
	Epsilon Decay	0.995
	Learning Rate	0.0001
	Batch Size	120
	Epochs	5000



(a) Privacy protection on the fruit dry weight.

(b) Privacy protection on indoor air temperature.

(c) Privacy protection on photosynthesis.

(d) Privacy protection on maintenance respiration.

Fig. 7. Use DQN model for managing the privacy protection level in a smart contract.

wind speed, sunlight intensity at the top of the canopy, light intensity from top lamps at the top of the canopy, light intensity from inter-lighting at the top of the canopy, indoor air temperature, indoor air humidity, indoor air carbon dioxide concentration, plant photosynthesis, plant maintenance respiration, plant growth respiration, plant fruit dry weight, plant leaf dry weight, plant stem dry weight. The DQN model has 7 output actions for tuning roof ventilation,

including keeping the same, increasing 0.1, increasing 0.3, increasing 0.5, decreasing 0.1, decreasing 0.3, and decreasing 0.5. The reward is defined by the growth weight of the crop. The other key parameters of the DQN model are shown in Table 8.

4.4.2 Testing Results. We use DQN as the control model to check whether the control data that will be shared in a smart contract could produce an amount of yield that is higher than the requirement of the data owner. We first use a greenhouse simulator GreenLight to train the DQN model. GreenLight can simulate a greenhouse with tomato growth under various outdoor and indoor environments, and it can simulate most of the important control approaches to greenhouses, such as irrigation, ventilation, and lighting. After training the DQN model, we input a series of control data together with related sensing data into the model. We assume the accumulation dry weight of the crop (WSO) is the aim to be protected, which means WSO cannot be higher than the preset value of the data owner. In the original data, there is a control value for every one hour. If WSO is higher than the preset value, we will reduce the size of control data that is input into the smart contract. For this purpose, we sample the control value every 6, 12, 18, and 24 hours respectively, and input the new sample control data into the smart contract. In the experiment, we also use the same approach to protect the indoor air temperature data.

The result is shown in Figure 7. In Figure 7(a), as the time interval increases from 6 hours to 24 hours, the WSO difference between the original control data and the data with the new control interval increases from 6% to 12%. When the time interval is 12 hours, it shows the highest yield difference. This is mainly because we only make the test in two days, which is too short to show a stable trend. After that, we tested the RMSE of indoor air temperature, photosynthesis, and maintenance respiration with the time interval increased from 6 hours to 24 hours. The results are shown in Figure 7(b), Figure 7(c), and Figure 7(d) respectively. As shown in the results, the trends of these testing results are the same, in which the testing results increase as the time interval increases. This means that the privacy protection level can be controlled by selecting a suitable subset of data, while the optimal subset can be evaluated and selected by the DRL model.

5 FUTURE WORK

5.1 Real-world Attacks on Private Data

Despite the data integrity protection that the blockchain offers, the decentralized nature of blockchain brings transparency issues, which pose threats to data privacy. Although the current Laplace mechanism satisfies DP and should provide privacy protection, a few issues persist. The main issue is that no real-world malicious privacy attacks are conducted to evaluate the system. For that reason, we will conduct additional research to discover potential privacy attacks.

5.2 Data Encoding Mechanism

The DP algorithm used in this work does not provide privacy protection for statistical information, as the algorithm only offers protection for each data entry. Therefore, additional privacy-preserving techniques could be applied to offer broader privacy protections. In addition, besides the local DP used in this paper, there is another DP called global DP. The Laplace mechanism in a global setting can be further investigated. Global DP requires a trusted third party to inject noise into the query output instead of individual data entries. A trusted third party means a centralized solution for data storage, such as a central server, instead of IPFS, which makes this solution provide a flexible utility privacy trade-off.

5.3 Management of Smart Contract

In our system, we leverage DRL to manage which part of control data can be exposed to data users, so that the optimal control method is protected. Although this method can protect the control data, there is still a risk that the data is exposed to users. For example, the users could create multiple contracts with the data owner. After obtaining the data, the user could merge the data to have a more complete data set. To cope with this issue, a user recognition component could be built in the future. Each contract should be checked whether a malicious user collects data by using multiple contracts. Meanwhile, if a user must extract control data multiple times, the data encoding solution could be used before using DRL to pick data.

5.4 Time Overhead on Using ML and DL Models

Our solution uses the ML and DL models to control the privacy protection level and uses the DRL model to manage smart contracts. These models are trained using historical data. Therefore, the time overhead running the modeling after deployment is very low. However, it is time-consuming to train the models before deployment. Especially, when the deployment scenarios change and new sensing data is collected, the models must be re-trained and re-calibrated to achieve optimal performance. In the future, we will research how to leverage transfer learning to allow models to cope with various scenarios and reduce the time overhead to train the new models. Meanwhile, we plan to build a DL-based approach to automatically calibrate the parameters of the models so that they can always have the optimal performance for various scenarios after training.

6 CONCLUSION

In this research, we propose a system to protect the integrity and privacy of multimodal data produced by IoT systems. The system can provide data integrity protection by utilizing IPFS and smart contracts. Meanwhile, the privacy protection level can be controlled by the results of ML models, and the smart contract could be further managed for authorized data users by the DRL model. We tested our system in extensive experiments, including DP protection, ML-based privacy protection level, data storage on the Ethereum blockchain, and the DRL model for managing smart contracts. The testing results show that our system can not only protect the privacy and integrity of IoT multimodal data, but also control the desired level of privacy and integrity protection. We believe the system shows great potential for privacy and integrity protection, which can be used as a guiding scheme for future research in this area.

REFERENCES

[1] Osama Elsherbiny, Lei Zhou, Yong He, and Zhengjun Qiu. A novel hybrid deep network for diagnosing water status in wheat crop using iot-based multimodal data. *Computers and Electronics in Agriculture*, 203:107453, 2022.

[2] Jianbang Dai, Xiaolong Xu, and Fu Xiao. Glads: A global-local attention data selection model for multimodal multitask encrypted traffic classification of iot. *Computer Networks*, 225:109652, 2023.

[3] Rahul Dagar, Subhranil Som, and Sunil Kumar Khatri. Smart farming – iot in agriculture. In *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*, pages 1052–1056, 2018.

[4] Elsayed Said Mohamed, AA Belal, Sameh Kotb Abd-Elmabod, Mohammed A El-Shirbeny, A Gad, and Mohamed B Zahran. Smart farming for improving agricultural management. *The Egyptian Journal of Remote Sensing and Space Science*, 2021.

[5] J Efrim Boritz. Is practitioners' views on core concepts of information integrity. *International Journal of Accounting Information Systems*, 6(4):260–279, 2005.

[6] Edoardo Gaetani, Leonardo Aniello, Roberto Baldoni, Federico Lombardi, Andrea Margheri, and Vladimiro Sassone. Blockchain-based database to ensure data integrity in cloud computing environments. In *ITASEC*, pages 146–155, 2017.

[7] Leanne Wiseman, Jay Sanderson, Airong Zhang, and Emma Jakku. Farmers and their data: An examination of farmers' reluctance to share their data through the lens of the laws impacting smart farming. *NJAS-Wageningen Journal of Life Sciences*, 90:100301, 2019.

[8] Shangping Wang, Yinglong Zhang, and Yaling Zhang. A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems. *Ieee Access*, 6:38437–38450, 2018.

[9] Nabil Rifi, Elie Rachkidi, Nazim Agoulmine, and Nada Chendeb Taher. Towards using blockchain technology for ehealth data access management. In *2017 fourth international conference on advances in biomedical engineering (ICABME)*, pages 1–4. IEEE, 2017.

[10] Ahmed El Ouadrhiri and Ahmed Abdelhadi. Differential privacy for deep and federated learning: A survey. *IEEE access*, 10:22359–22380, 2022.

[11] Ahmed Afif Monrat, Olov Schelén, and Karl Andersson. A survey of blockchain from the perspectives of applications, challenges, and opportunities. *IEEE Access*, 7:117134–117151, 2019.

[12] Nouhaila El Akrami, Mohamed Hanine, Emmanuel Soriano Flores, Daniel Gavilanes Aray, and Imran Ashraf. Unleashing the potential of blockchain and machine learning: Insights and emerging trends from bibliometric analysis. *IEEE Access*, 2023.

[13] Arunima Ghosh, Shashank Gupta, Amit Dua, and Neeraj Kumar. Security of cryptocurrencies in blockchain technology: State-of-art, challenges and future prospects. *Journal of Network and Computer Applications*, 163:102635, 2020.

[14] Dejan Vujičić, Dijana Jagodić, and Siniša Ranđić. Blockchain technology, bitcoin, and ethereum: A brief overview. In *2018 17th international symposium infoteh-jahorina (infoteh)*, pages 1–6. IEEE, 2018.

[15] Satpal Singh Kushwaha, Sandeep Joshi, Dilbag Singh, Manjit Kaur, and Heung-No Lee. Systematic review of security vulnerabilities in ethereum blockchain smart contract. *IEEE Access*, 10:6605–6621, 2022.

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

[17] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

[18] Ying Zhao and Jinjun Chen. A survey on differential privacy for unstructured data content. *ACM Computing Surveys (CSUR)*, 54(10s):1–28, 2022.

[19] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

[20] Quan Geng and Pramod Viswanath. The optimal noise-adding mechanism in differential privacy. *IEEE Transactions on Information Theory*, 62(2):925–951, 2015.

[21] Ya-Nan Cao, Yujue Wang, Yong Ding, Zhenwei Guo, Qianhong Wu, and Hai Liang. Blockchain-empowered security and privacy protection technologies for smart grid. *Computer Standards & Interfaces*, page 103708, 2022.

[22] Baodong Wen, Yujue Wang, Yong Ding, Haibin Zheng, Bo Qin, and Changsong Yang. Security and privacy protection technologies in securing blockchain applications. *Information Sciences*, page 119322, 2023.

[23] Li Duan, Wenyao Xu, Wei Ni, and Wei Wang. Bsaf: A blockchain-based secure access framework with privacy protection for cloud-device service collaborations. *Journal of Systems Architecture*, 140:102897, 2023.

[24] Jingwei Liu, Xiaolu Li, Lin Ye, Hongli Zhang, Xiaojiang Du, and Mohsen Guizani. Bpds: A blockchain based privacy-preserving data sharing for electronic medical records. In *2018 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE, 2018.

[25] Caixia Yang, Liang Tan, Na Shi, Bolei Xu, Yang Cao, and Keping Yu. Authprivacychain: A blockchain-based access control framework with privacy protection in cloud. *IEEE Access*, pages 70604 – 70615, 2020.

[26] Xinyan Li, Huimin Zhao, and Wu Deng. Bfod: Blockchain-based privacy protection and security sharing scheme of flight operation data. *IEEE Internet of Things Journal*, 2023.

[27] Rathindra Sarathy and Krishnamurty Muralidhar. Evaluating laplace noise addition to satisfy differential privacy for numeric data. *Trans. Data Priv.*, 4(1):1–17, 2011.

[28] Zihao Shan, Kui Ren, Marina Blanton, and Cong Wang. Practical secure computation outsourcing: A survey. *ACM Computing Surveys (CSUR)*, 51(2):1–40, 2018.

[29] Jaewoo Lee and Chris Clifton. Differential identifiability. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1041–1049, 2012.

[30] Ninghui Li, Wahbeh Qardaji, Dong Su, Yi Wu, and Weining Yang. Membership privacy: A unifying framework for privacy definitions. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 889–900, 2013.

[31] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International journal of uncertainty, fuzziness and knowledge-based systems*, 10(05):557–570, 2002.

[32] Junbo Zhang, Yu Zheng, Dekang Qi, Ruiyuan Li, and Xiuwen Yi. Dnn-based prediction model for spatio-temporal data. In *Proceedings of the 24th ACM SIGSPATIAL international conference on advances in geographic information systems*, pages 1–4, 2016.

[33] Qingzhi Liu, Tiancong Xia, Long Cheng, Merijn Van Eijk, Tanir Ozcelebi, and Ying Mao. Deep reinforcement learning for load-balancing aware network control in iot edge systems. *IEEE Transactions on Parallel and Distributed Systems*, 33(6):1491–1502, 2021.

[34] Qingzhi Liu, Long Cheng, Adele Lu Jia, and Cong Liu. Deep reinforcement learning for communication flow control in wireless mesh networks. *IEEE Network*, 35(2):112–119, 2021.

[35] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[36] Jianbo Du, Wenjie Cheng, Guangyue Lu, Haotong Cao, Xiaoli Chu, Zhicai Zhang, and Junxuan Wang. Resource pricing and allocation in mec enabled blockchain systems: An a3c deep reinforcement learning approach. *IEEE Transactions on Network Science and Engineering*, 9(1):33–44, 2021.

[37] M Mehdi Afsar, Trafford Crump, and Behrouz Far. Reinforcement learning based recommender systems: A survey. *ACM Computing Surveys*, 55(7):1–38, 2022.

[38] Yan Ge and Haixia Wu. Prediction of corn price fluctuation based on multiple linear regression analysis model under big data. *Neural Computing and Applications*, 32:16843–16855, 2020.

[39] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[40] David Katzin, Simon van Mourik, Frank Kempkes, and Eldert J van Henten. Greenlight–an open source model for greenhouses with supplemental lighting: Evaluation of heat requirements under led and hps lamps. *Biosystems Engineering*, 194:61–81, 2020.