Hindawi
Security and Communication Networks
Volume 2020, Article ID 2017930, 16 pages
https://doi.org/10.1155/2020/2017930

Research Article

Protection of Sensitive Data in Industrial Internet Based on Three-Layer Local/Fog/Cloud Storage

Jing Liu[1,2] **Changbo Yuan**,[1] **Yingxu Lai**[1,3] **and Hua Qin**[4]

[1]*College of Computer Science, Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China*
[2]*Shaanxi Key Laboratory of Network and System Security, Xidian University, Xian 710071, China*
[3]*Science and Technology on Information Assurance Laboratory, Beijing 100072, China*
[4]*Information Technology Support Center, Beijing University of Technology, Beijing 100124, China*

Correspondence should be addressed to Yingxu Lai; laiyingxu@bjut.edu.cn

Received 27 October 2019; Revised 3 February 2020; Accepted 5 February 2020; Published 2 April 2020

Academic Editor: Prosanta Gope

Industrial Internet technology has developed rapidly, and the security of industrial data has received much attention. At present, industrial enterprises lack a safe and professional data security system. Thus, industries urgently need a complete and effective data protection scheme. This study develops a three-layer framework with local/fog/cloud storage for protecting sensitive industrial data and defines a threat model. For real-time sensitive industrial data, we use the improved local differential privacy algorithm M-RAPPOR to perturb sensitive information. We encode the desensitized data using Reed–Solomon (RS) encoding and then store them in local equipment to realize low cost, high efficiency, and intelligent data protection. For non-real-time sensitive industrial data, we adopt a cloud-fog collaborative storage scheme based on AES-RS encoding to invisibly provide multilayer protection. We adopt the optimal solution of distributed storage in local equipment and the cloud-fog collaborative storage scheme in fog nodes and cloud nodes to alleviate the storage pressure on local equipment and to improve security and recoverability. According to the defined threat model, we conduct a security analysis and prove that the proposed scheme can provide stronger data protection for sensitive data. Compared with traditional methods, this approach strengthens the protection of sensitive information and ensures real-time continuity of open data sharing. Finally, the feasibility of our scheme is validated through experimental evaluation.

1. Introduction

Intelligent manufacturing, which consists of a man-machine integrated intelligent system with intelligent machines and human experts, is an inevitable trend in the continuing development of the global manufacturing industry [1]. In recent years, with the rapid development of Industrial Internet technology, industrial data protection has attracted much research interest [2]. In the industrial production process, a large amount of sensitive data is generated, including data from the manufacturing process of the production line, product cost information, operations data, operations information, marketing strategy, intellectual property rights, and customer data. If this sensitive data is leaked, it may lead to significant business information loss or even affect the reputation of an enterprise. In recent years, information leakage incidents have occurred repeatedly. For example, in 2017, Equifax (Atlanta, US) announced that a data breach had occurred. Recently, UpGuard (Sydney, Australia), a cybersecurity company, reported that researchers found more than 540 million records on Amazon's S3 server, including Facebook user information such as comments, responses, and account names [3]. Therefore, in the context of the Industrial Internet, it is a major challenge to ensure that sensitive information is not leaked.

In actual industrial scenarios, some industrial data must be processed in real time, such as predictive maintenance data in current intelligent factories. If computer numerical control (CNC) machine tools fail, delays in the product production cycle as well as economic losses could occur. Therefore, to avoid equipment failure, factories should perform predictive maintenance work such as collecting

sensor data in real time, predicting possible situations, and adjusting the equipment according to the predicted situation in a timely manner. Sensor information such as node locations and the spatial coordinates of the sensors themselves also constitute sensitive data. We label this as real-time sensitive data. To prevent the leakage of sensitive information and to ensure data security, a secure storage method is needed to store real-time sensitive data, including equipment locations, safely and without affecting data availability.

In addition, a large amount of sensitive data is not processed in real time, including a company's financial data, inventory data, production data, marketing plans, customer information, intellectual property rights, and supplier information [4]. We label this as non-real-time sensitive data. This data is usually stored with third-party cloud service providers who may claim that the data is encrypted but cannot verify the encryption. To prevent data leakage, we cannot store the non-real-time sensitive data in the cloud. Thus, for non-real-time sensitive data, it is necessary to design a data protection scheme that can fully utilize cloud storage and ensure data security.

The recent development of cloud-fog collaborative computing provides a new approach to solve this problem. Cloud computing combines relatively independent computing technology and network technology, superimposes distributed computing power on a converged network platform, and effectively integrates network and computing resources by virtualization technology, which is a major breakthrough in IT technology. Fog computing is an extension and a powerful complement of cloud computing. Figure 1 shows the architecture of cloud-fog computing. It includes the factory terminal equipment layer, fog computing layer, and cloud computing layer. The main task of the factory terminal equipment layer is to collect data and upload it to the fog. The fog computing layer, the middle layer in this architecture, plays an important role between the cloud computing layer and the fog nodes. The fog nodes have a given storage capacity and computing capacity. The introduction of fog computing can reduce the cloud computing layer and improve work efficiency. The cloud computing layer has a high storage capacity and computing power. Because non-real-time processed data cannot be completely stored in the cloud, we adopt the cloud-fog collaborative storage scheme. In this study, cloud-fog collaborative storage means that the cloud nodes and the fog nodes store corresponding data according to the storage strategy; that is, some data is stored in the fog nodes and other data is stored in the cloud nodes.

To solve the problem of secure storage for real-time and non-real-time sensitive industrial data, this study designs a three-layer protection framework with local/fog/cloud storage for sensitive industrial data. For real-time sensitive industrial data, we use the improved local differential privacy algorithm M-RAPPOR to perturb sensitive information (location information, etc.) and then encode the masked data in local equipment. In addition, we adopt the optimal scheme of distributed storage in local equipment to realize low-cost, high-efficiency data protection. For non-real-time sensitive industrial data, we adopt cloud-fog collaborative storage to achieve multilevel protection. This framework greatly improves security and restorability and relieves the storage pressure of local equipment.

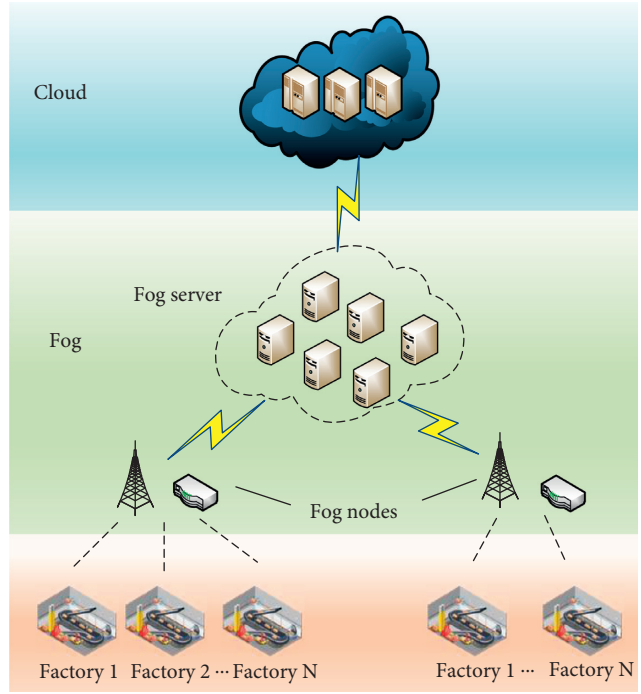

Figure 1: Architecture of cloud-fog computing.

The remainder of this paper is organized as follows. Section 2 reviews related works. Section 3 describes the protection framework for sensitive data and defines the threat model. Section 4 presents experiments for two different schemes and evaluates the results through several indicators. Finally, Section 5 presents our conclusions.

2. Related Work

With the continuous developments in information technology in the era of big data, the problem of data security has attracted increasing attention. How to ensure that sensitive data is not leaked is a major challenge at present. Cloud-fog collaboration is a new solution to meet the current needs of Internet of Things (IoT) applications. However, there is a risk of sensitive-data leakage in both cloud computing and fog computing. At present, therefore, data security has emerged as an important issue for both academia and industry. As a result, experts and researchers have proposed their own opinions and data protection programs. This section introduces the current research results from two aspects—the protection of sensitive data and cloud-fog collaborative computing.

2.1. Protection of Sensitive Data

2.1.1. Protection of Sensitive Data Based on Differential Privacy. Dwork and Lei [5] proposed differential privacy techniques as a privacy protection model that defines an

extremely rigorous attack model that operates without regard on how much background knowledge the attacker has. Kenthapadi et al. [6] used differential privacy to address sensitive-data protection issues for collective user behavior. Mohammed et al. [7] proposed a differential privacy algorithm for private data distribution and vertical partition data. In a scenario where third-party data collectors are not trusted, Duchi et al. [8] proposed local differential privacy, where every user understands their privacy protection process and can individually process and protect personal sensitive information. Erlingsson [9] developed the Randomized Aggregatable Privacy-Preserving Ordinal Response (RAPPOR) privacy algorithm that enables Google to collect user answers to questions such as a browser's default home page and default search engine to understand the unwelcome or malicious hijacking of user settings. Gu et al. [10] proposed a privacy protection mining (DIFF-PPM) algorithm combined with the Markov chain Monte Carlo algorithm to provide privacy protection and maintain data availability when the (ε, δ)-differential privacy conditions are met.

Most of these studies have aimed to protect users' personal sensitive data, and they have not applied differential privacy to the industrial field. In addition, the weakness of differential privacy is obvious: because the assumptions about background knowledge are too strong, a large amount of randomization needs to be added to the query results, resulting in a sharp decline in data availability. Therefore, how to balance the level of sensitive-data protection and data availability according to specific application scenarios remains unclear and challenging in research on differential privacy applications.

2.1.2. Sensitive Data Protection Based on Fog Computing. Xu et al. [11] introduced local differential privacy into fog computing and proposed a framework for local differential privacy protection; however, they did not mention specific implementation details and privacy algorithms. Wang et al. [12] proposed a three-layer privacy protection framework; however, they did not consider the problem of local equipment failure, in which case users would not be able to query complete data. Du et al. [13] proposed a fog computing support data center query model based on differential privacy and proved, through rigorous mathematical deduction, that it ensured the reliability and effectiveness of privacy protection. Lyu et al. [14] proposed the PPFA privacy protection aggregation system that uses the stability of a Gauss mechanism to ensure the differential privacy of the statistical results and reduces the loss of privacy by combining a stream cipher with a public key cipher to maintain practicability. Huang et al. [15] proposed an attribute-based encryption scheme that makes full use of fog servers. The collected data is encrypted by fog servers and then outsourced to cloud servers. The experimental results show that the functions of the scheme are limited because it does not support efficient data search, and the fog server takes up a large part of the workload. Lu et al. [16] proposed the Lightweight Privacy-preserving Data Aggregation (LPDA) scheme. Experiments indicated that this scheme was highly efficient in terms of computational cost and communication overhead. Wang et al. [17] proposed a fog computing query model based on differential privacy. Compared with the traditional privacy protection model, this model affords different degrees of improvement in computational overhead, execution efficiency, and energy consumption. Kulkarni et al. [18] proposed a privacy protection framework to protect the data security of sensors between the terminal equipment and fog networks. This scheme introduced a selective public key cryptosystem that can resist internal attacks. However, they did not consider the computational overhead, and the fog nodes may overload and collapse if the sensor transmits data for a long time.

The above studies introduced fog computing to protect sensitive data; however, most of them ignored the problem of the limited computational power and storage capacity of fog nodes. Many data protection schemes based on fog computing do a lot of work on the fog side; this directly leads to increased computational overhead and energy consumption, resulting in jitter or even downtime of fog nodes.

2.1.3. Protection of Sensitive Data in Cloud Computing. In recent years, data security in cloud storage has attracted attention in industry and academia. Many experts and researchers have proposed schemes to address this issue. Hou et al. [19] proposed protection schemes that use encrypted data by the system when data has been transferred by SSL. Feng [20] proposed a method in which data is encrypted in a closed cloud environment to solve the problem of a data leakage caused by the increased burden of the cloud server. In addition, one-time encryption and multipoint secure storage can be realized. However, encryption makes it more difficult to search in the cloud. At present, a key issue in the field of cloud computing is searchable encryption. Fu et al. [21–23] and Xia et al. [24] provided different solutions to this problem that improved accuracy, safety, and efficiency. Kulkarni et al. [25] designed a virtual private storage service based on a newly developed encryption technology. This service achieved the best combination of private cloud security and public cloud functionality. Wang et al. [26] proposed that users do not have actual ownership of outsourced data; this makes data integrity protection in cloud computing a difficult task. Shen et al. [27] proposed an efficient public auditing protocol that includes a doubly-linked information table and a location array. Experiments show that this protocol can reduce computational and communication overheads and achieve certain efficiencies.

The above studies improved the use of cloud storage for the protection of sensitive data. However, they suffer from a common problem: they cannot provide a defense against internal attacks from the cloud.

2.2. Cloud-Fog Collaborative Computing. At present, the academic community mainly solves the problem of task scheduling through cloud-fog coordination. Pham and Huh. [28] considered task scheduling in the cloud-fog collaboration computing system; they proposed a task scheduling

algorithm that guarantees the performance of application execution. However, they did not consider the fog provider's budget. Deng et al. [29] studied the balance between power consumption and delay in cloud-fog collaborative computing systems. They first modeled the power consumption and delay functions of various parts of the system and formulated allocation problems according to the workload. Then, they developed an approximate solution to the original problem by decomposition and separately formulated three subproblems of the corresponding subsystem. Experiments showed that fog computing could reduce communication delay and significantly improve cloud computing performance. Bierzynski et al. [30] believed that cloud-fog collaboration can solve most problems in IoT and discussed four possible ways to distribute workloads between different levels.

The above studies solved the task scheduling problem through cloud-fog collaboration; however, they did not consider network security and data protection issues. In addition, simply relying on fog computing or cloud computing cannot provide more secure protection for industrial data.

In summary, the studies discussed in this section solved the data protection problem and task scheduling problem to some extent. However, for industrial applications with both a large amount of data and a high degree of data protection, their results cannot meet the requirements of the Industrial Internet update speed, high data accuracy, and large data volume. Thus far, academic and industrial research on the protection of sensitive data in the Industrial Internet of Things (IIoT) remains in its infancy. In this study, we propose a protection model for sensitive data that provides higher protection and is more adaptable to industrial environments.

3. Sensitive Industrial Data Protection Scheme

Section 3.1 introduces our protection framework for sensitive industrial data and the threat model. Sections 3.2 and 3.3 describe the protection of data for real-time and non-real-time processing, respectively.

3.1. Our Work and Contribution

3.1.1. Sensitive Industrial Data Protection Framework. This study designs a three-layer protection framework with local/fog/cloud storage for sensitive industrial data, as shown in Figure 2. This framework mainly consists of three modules: M-RAPPOR disturbance module, Reed–Solomon (RS) encoding module, and Advanced Encryption Standard (AES) encryption and RS encoding module. This framework not only achieves low-cost, high-efficiency, and intelligent data protection but also alleviates the storage pressure of local equipment through improved recoverability.

We divide industrial data into real-time and non-real-time sensitive data depending on its characteristics.

(1) For real-time sensitive data, we design a data protection scheme based on local differential privacy and RS encoding. We apply the M-RAPPOR algorithm to the sensitive data (location information), apply RS encoding to the perturbed data and the undisturbed nonsensitive data, and store them in local equipment. Considering the storage capacity of local equipment and the problem that data cannot be recovered owing to local equipment failures, we adopt the optimal solution of distributed storage in the local equipment and add corresponding restrictions to the RS encoding, which not only solves the data recovery problem due to the failure of local equipment but also improves the encoding efficiency and reduces the computational cost. If we need to perform predictive maintenance analysis, we can decode the data and then analyze it.

(2) For non-real-time sensitive data, we design a data protection scheme based on AES encryption and RS encoding. The local equipment encrypts the non-real-time data using AES, uploads the ciphertext to the fog nodes, and performs RS encoding of the data on the fog nodes. Next, the encoded data is stored in the fog nodes and the cloud nodes according to the cloud-fog collaborative storage strategy. In this way, if an attempt is made to steal the data from the fog nodes or the cloud nodes, all the data cannot be taken.

3.1.2. Threat Model. Figure 3 shows the threat model that is defined according to the data protection framework.

There are two threats to industrial data:

(1) For real-time sensitive data, attackers can attack the local data center. They can steal location information of equipment to infer the general plant layout. The layout is a type of sensitive industrial data, and therefore, this attack will lead to the leakage of such data.

(2) For non-real-time sensitive data, attackers (internal personnel of the third-party) can abuse data mining to infer customers' private information and the factory's business strategy from the factory's customer information and marketing plans, respectively. This will not only threaten customers' privacy but also affect the factory's overall interests.

3.2. Protection of Real-Time Industrial Data

3.2.1. Local Differential Privacy. Local differential privacy considers scenarios in which third-party data collectors are untrustworthy. Each user perturbs the data according to the privacy algorithm and then uploads the perturbed data to the data collector. Local differential privacy is formally defined as follows.

Definition 1. There are n users, where each user corresponds to a record. A privacy algorithm M is given having a domain $\mathrm{Dom}(M)$ and a range $\mathrm{Ran}(M)$. If algorithm M obtains the same output result T ($T \subseteq \mathrm{Ran}(M)$) on any two records t and

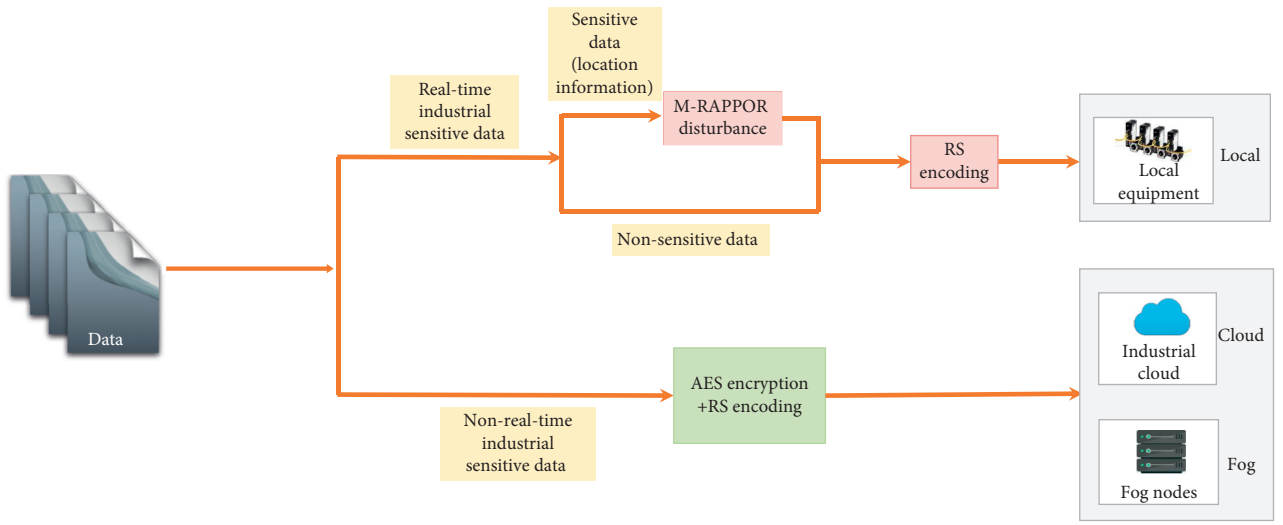

Figure 2: Protection framework for sensitive industrial data.

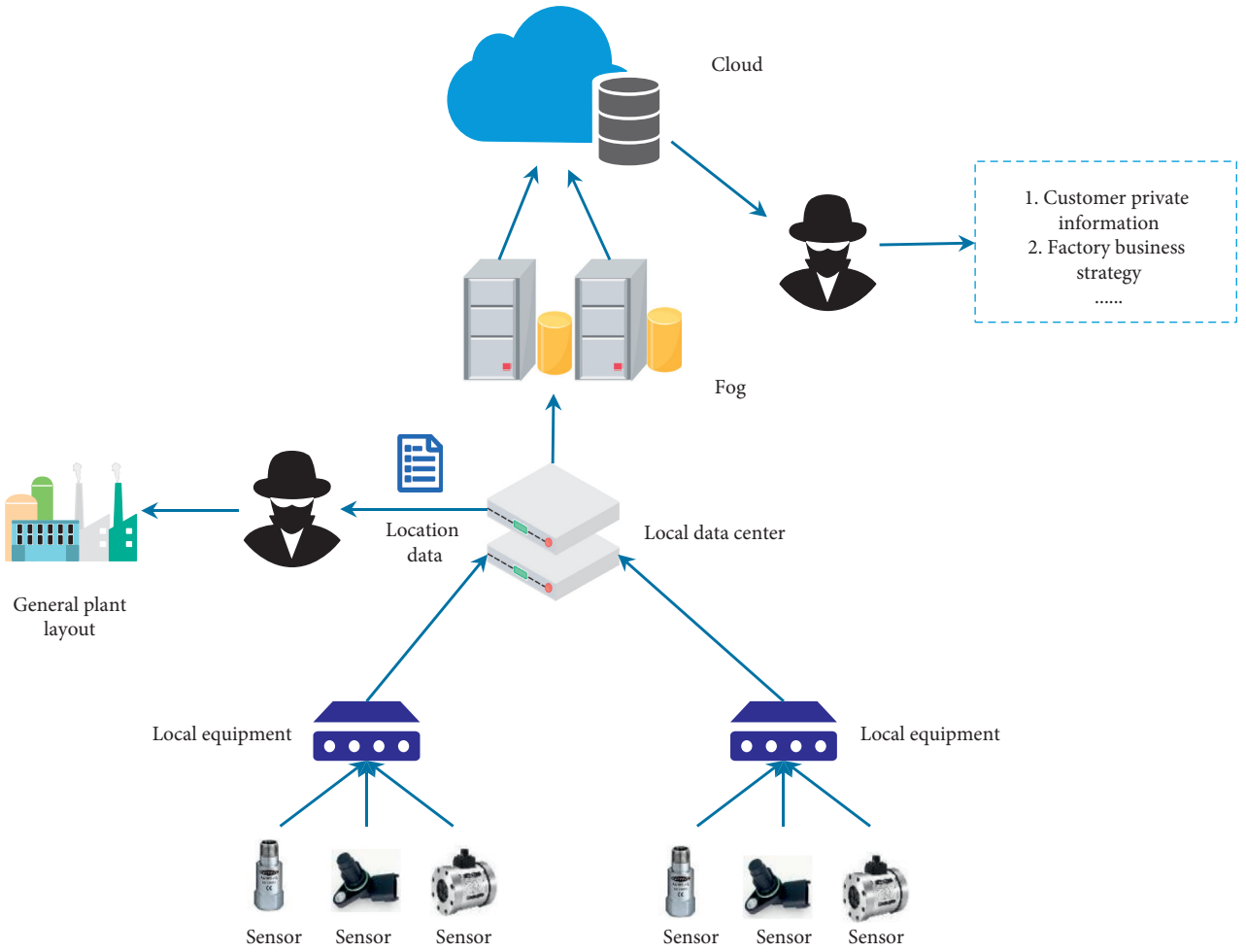

Figure 3: Threat model of sensitive industrial data.

$t'(t, t' \in \mathrm{Dom}(M))$ and satisfies the following inequalities, then M satisfies local differential privacy:

$$\Pr[M(t) = T] \leq e^{\varepsilon} \times \Pr[M(t') = T]. \quad (1)$$

As can be seen from Definition 1, the local differential privacy ensures that the algorithm M satisfies the ε-local differential privacy by controlling the similarity of the output results of any two records. In short, according to the

output result of the privacy algorithm M, it is almost impossible to infer which record the input data is. With local differential privacy technology, each user can process individual data independently. In other words, the process of privacy processing is transferred from the data collector to a single user, and therefore, there is no need for trusted third-party intervention. This can avoid the attacks that may be brought on by untrusted third-party data collectors. The implementation of local differential privacy protection requires the intervention of a data disturbance mechanism. Random response technology is the mainstream disturbance mechanism for local differential privacy protection technology.

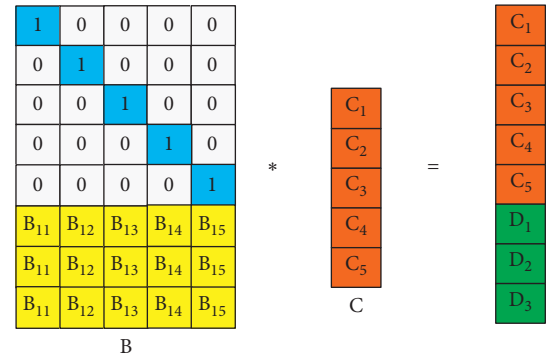

Figure 4: RS encoding process.

3.2.2. Reed–Solomon Encoding. RS encoding is a matrix operation. As shown in Figure 4, the input data is treated as matrix $C = (C_1, C_2, \ldots, C_n)$, and the encoding matrix B is multiplied by matrix C to obtain data blocks $(C_1, C_2, \ldots, C_n)$ and y redundant data blocks ($n = 5$, $y = 3$). The encoding matrix B must have reversibility of any submatrix. After RS encoding, the data is divided into n parts and y redundant data are generated. In the data of these $n + y$ parts, one can recover the total data with at least n data values. In other words, the total data cannot be recovered with less than n data values.

3.2.3. Data Protection Scheme Based on Local Differential Privacy and Reed–Solomon Encoding. With continuous developments in industries, many factories have started using equipment such as CNC machine tools to increase productivity. However, some problems have also been exposed. For example, a serious failure of a CNC machine tool will not only lead to the shutdown of the entire production line but also seriously delay the product production cycle. Therefore, to avoid this situation, the factory should analyze the data collected by sensors, predict possible situations, and warn operators or adjust the equipment according to the forecasted situation in a timely manner to avoid an accidental shutdown of the equipment. However, the data contains not only real-time data collected by the sensor but also a large amount of sensitive data of the factory equipment, such as node location and spatial coordinates. This type of sensitive data related to locations should not be leaked before predictive maintenance analysis. Therefore, it needs to be desensitized before data analysis to eliminate the correspondence between the equipment and the location. In addition, because the real-time data information collected by factory sensors is of great value and high confidentiality, we cannot upload this data to the fog or the cloud; it must be stored in local equipment. The protection scheme for real-time sensitive data is described in detail below.

We use local differential privacy to desensitize real-time sensitive data. Then, we encode the perturbed data and undisturbed nonsensitive data by RS encoding and store it in local equipment. Figure 5 shows the protection process for real-time data.

The real-time data collected by the factory sensors is first uploaded to the local equipment. Then, the following process is performed in the local equipment:

(1) Data Disturbance with Local Differential Privacy. Because of the limited computing power of local equipment, it cannot perform complex computing tasks. Therefore, it is very important to design a lightweight privacy protection algorithm. Local differential privacy considers the untrustworthiness of the data collector, and its disturbance mechanism mainly includes a random response, information compression, and distortion. The disturbance framework of the random response technology is simple and intuitive, and the disturbance level can be directly quantified. Therefore, we choose local differential privacy to protect the sensitive data. We refer to the basic idea of the RAPPOR algorithm, which is an open-source project of Google, and improve RAPPOR to reduce the computational cost of local equipment while ensuring privacy and data availability. In this paper, we label the improved algorithm as M-RAPPOR, and we use it to disturb the location information. Section 3.2.5 discusses the specific disturbance steps.

(2) RS Encoding and Storage. We apply RS encoding to the perturbed data and undisturbed nonsensitive data and store the data in local equipment. However, owing to the continuous production of data in factories and the limited storage capacity of local equipment, the storage pressure on equipment is gradually increasing. In addition, if the local equipment fails unexpectedly, it will not be possible to completely restore the data, resulting in serious losses to the factory. To solve these two problems, we adopt the optimal solution, namely, distributed storage in the local equipment, and add corresponding restrictions to RS. This not only solves the problem of the high storage pressure on local equipment and the fact that data cannot be recovered owing to the failure of the local equipment but also improves the encoding efficiency and reduces the computational cost. Section 3.2.4 discusses the specific distributed storage strategy.

(3) RS Decoding and Analysis. When predictive maintenance analysis is performed, the data stored in the local equipment are decoded, and the analysis is performed according to the statistical location distribution and data. Section 3.2.5 discusses the specific statistical process.

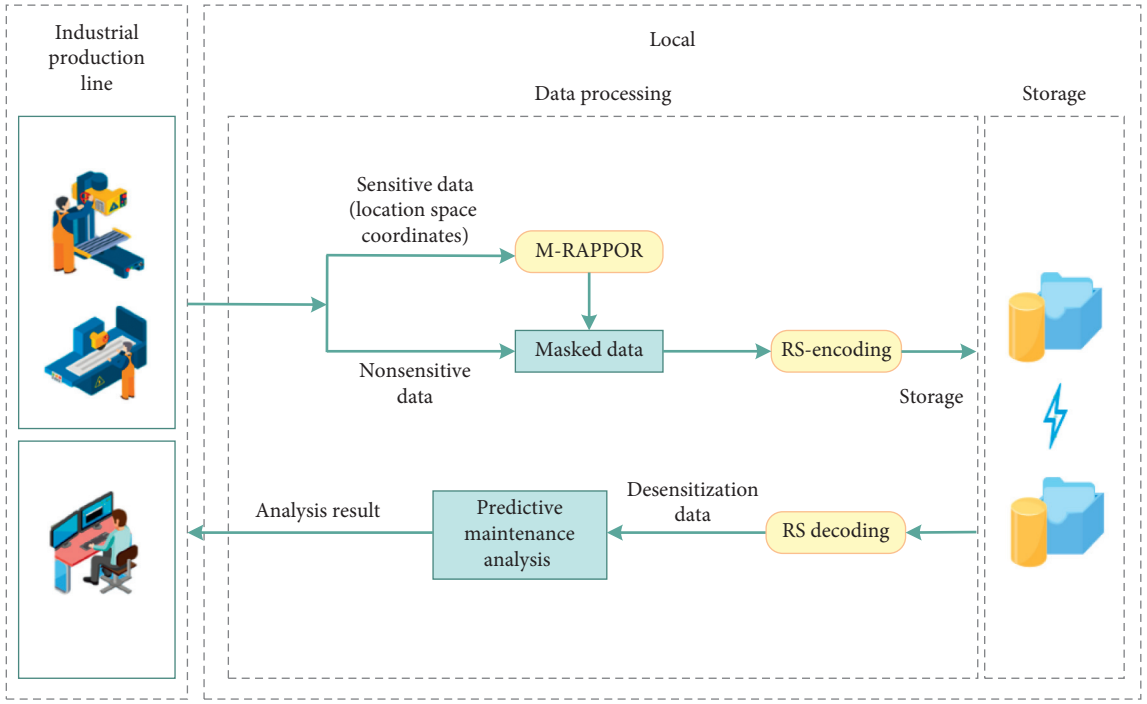

Figure 5: Protection process for real-time data.

3.2.4. Optimal Solution for Distributed Storage. In the previous section, we mentioned that distributed storage can solve the problem of high storage pressure on local equipment. However, to solve the problem in which the data cannot be completely recovered owing to the failure of the local equipment, it is necessary to impose restrictions on the redundant data blocks according to the characteristics of the RS encoding to ensure that the complete data can be recovered when any of the local equipment fails. Assuming that the encoded real-time data is b, the redundant data block is m. We store data block b in different local pieces of equipment according to the amount of local equipment. The storage capacity of local equipment is different for various industrial environments. Therefore, we assume that the storage capacities of equipment 1, equipment 2, ..., equipment n are $M_1, M_2, \ldots, M_n$, respectively; therefore, the data Z_i stored by equipment i is

$$Z_i = \frac{b \times M_i}{M_1 + M_2 + \cdots + M_n}. \tag{2}$$

To ensure that complete data can still be recovered when any local equipment fails, the redundant data m must satisfy the following inequality:

$$\frac{b \times \max\{M_1, M_2, \ldots, M_n\}}{M_1 + M_2 + \cdots + M_n} \le m < b. \tag{3}$$

In addition, with an increase in the number of redundant data blocks, the coding and decoding efficiency will decrease, as confirmed in Section 4. Therefore, m should be the minimum value; that is,

$$m = \frac{b \times \max\{M_1, M_2, \ldots, M_n\}}{M_1 + M_2 + \cdots + M_n}. \tag{4}$$

3.2.5. M-RAPPOR. The RAPPOR algorithm [9] is divided into two parts for the privacy protection of user data: data sender and data collector. In the predictive maintenance scenario considered in this study, the data sender is the local equipment that collects sensor data, and the data collector is the local data center that performs the predictive maintenance analysis. However, the processing capacity of local equipment is limited, and the location distribution after statistics is needed in predictive maintenance analysis; therefore, the computational cost of local equipment should be reduced while ensuring privacy and data availability.

This study proposes an improved M-RAPPOR algorithm based on RAPPOR. The following improvements are made:

(1) Data is disturbed only once; this can decrease the computational cost of the data sender. Because only one disturbance occurs, the level of privacy protection is decreased. Therefore, we add a probability factor to the permanent random response (PRR), and we use two probability factors to expand the flexibility of parameter adjustment so as to increase the level of privacy protection.

(2) The number of hash functions h is set to 1, and k of the Bloom filter is set to be equal to two times the number of data attribute values (increased scalability). After correction, the number of 1 on each bit of the Bloom filter is the number of corresponding

attribute values, the lasso regression is not needed, and therefore, the computation cost of the data collector is decreased.

The M-RAPPOR algorithm process is divided into the data sender (local equipment) and the data collector (local data center).

(1) Data Sender

(1) The length of the bit array B is set to k, and the number h of the hash function is set to 1. The initial value of all bits is set to 0. The sensitive data v is hashed only once on the bit array B, and the bit value corresponding to the hash value is changed from 0 to 1.

(2) The PRR is obtained by perturbing each bit of bit array B using a random response technique. The disturbance is performed according to the following equation. We introduce the second probability factor f_2, where $f_1, f_2 \in [0, 1]$ represents the probability value:

$$P\left(B_i' = x\right) = \begin{cases} f_1, x = 1, \\ f_2, x = 0, \\ 1 - f_1 - f_2, x = B_i. \end{cases} \tag{5}$$

To satisfy ε-local differential privacy, the privacy budget ε is calculated:

$$\varepsilon = \ln \frac{1 - f_2}{f_1}. \tag{6}$$

Compared with RAPPOR, the computational cost is decreased because the data is only mapped using the Bloom filter and is disturbed once.

(2) Data Collector

(1) The data collector collects all vectors B' sent by the data sender and counts the number D_i of 1 values corresponding to each bit i on B'.

(2) Each bit D_i is corrected, where N is the total number of data that is sent by the data sender. The corrected results n_i are expressed as follows:

$$n_i = \frac{N\left(f_2 - 2f_1 f_2\right)}{2f_1 - 4f_1 f_2 - 1 + 2f_2} + \frac{D_i\left(f_1 - 1 + f_2\right)}{2f_1 - 4f_1 f_2 - 1 + 2f_2}. \tag{7}$$

(3) The corresponding n_i is determined according to the hash map of the data attribute value. n_i is the frequency statistic of the attribute value.

In the algorithm of the data collector, lasso regression is not conducted; therefore, the computational cost is decreased.

3.2.6. Security Analysis. This section describes the security analysis of the proposed threat model and proves that the proposed security storage scheme can really protect sensitive industrial data.

According to the threat model proposed in Section 3.1.2, for real-time industrial data (predictive maintenance data), attackers will attack the local data center and infer the factory's internal framework map based on the obtained location data of the factory equipment; this will cause economic losses to the factory. When the proposed data security storage scheme is adopted, it is impossible for attackers to obtain specific location information for the following reasons:

(1) We assume that the data center is attacked and that the predictive maintenance data including location information is stolen. Because the location information is disturbed by M-RAPPOR on the local equipment, the attacker obtains the location information as a "0-1" string instead of specific location information.

(2) From the algorithm of the data collector, as described in Section 3.2.4, to obtain the frequency distribution of each location value, the list of candidate attribute values is needed; that is, it is necessary to know what the location attribute values are. It is difficult for attackers to collect all attribute value lists.

(3) We assume that attackers are intelligent enough to collect a list of all attribute values. According to equation (7), the corrected result n_i is directly related to f_1 and f_2. However, f_1 and f_2 are set by us, and therefore, it is difficult for attackers to infer the corrected result n_i.

(4) We assume that attackers collect a list of all attribute values and infer f_1 and f_2. They will only obtain the frequency statistic of each location value, and they cannot obtain the corresponding relationship between the factory equipment and the location.

In summary, it is impossible for attackers to obtain the specific location information of the factory equipment. Therefore, the proposed scheme can effectively protect sensitive industrial data.

3.3. Protection of Non-Real-Time Industrial Data. Industrial data includes data processed in real time as well as data that is not processed in real time. Because of the large amount of non-real-time data, it is usually stored with third-party cloud service providers. However, relying on cloud encrypted storage alone cannot defend against internal attacks from the cloud; therefore, it cannot effectively solve the data security problem. Therefore, we design a data protection scheme with AES encryption and RS encoding for cloud-fog collaborative storage. According to the cloud-fog collaborative storage strategy, some encoded data is stored in the fog nodes and the rest of the data is stored in the cloud nodes.

3.3.1. Data Protection Scheme for Cloud-Fog Collaborative Storage. Figure 6 shows the proposed data protection scheme for cloud-fog collaborative storage of data that is not processed in real time. The scheme includes three layers—cloud nodes, fog nodes, and local equipment.

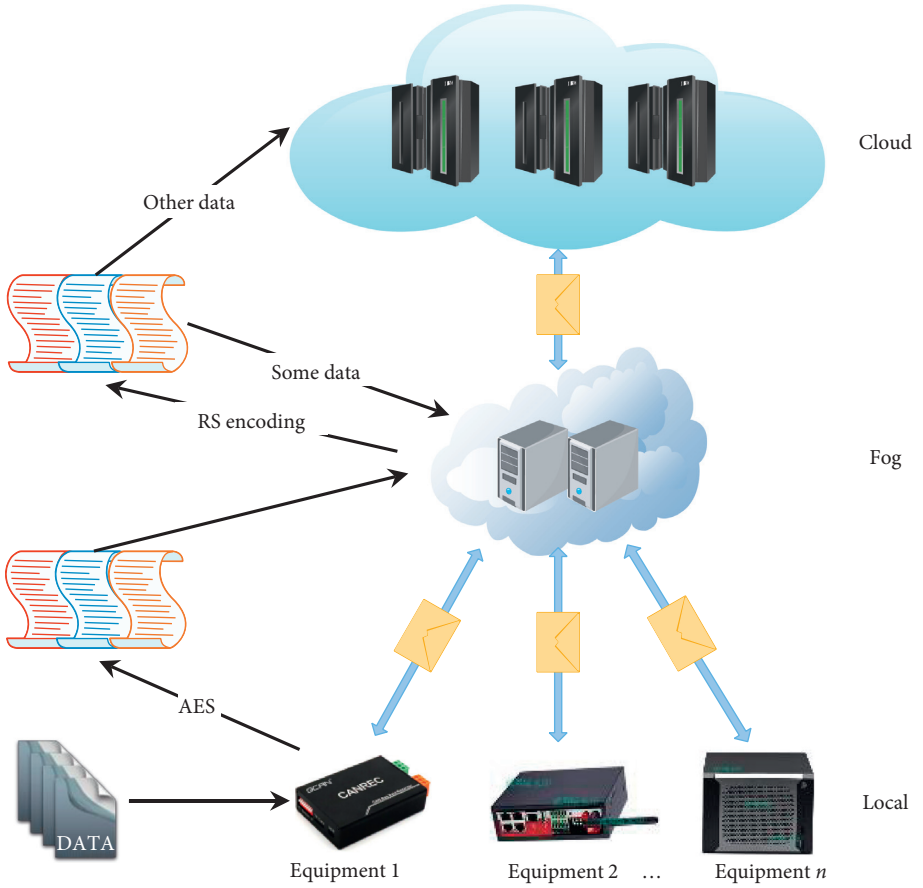

Figure 6: Protection process for data that does not require real-time processing.

The local equipment encrypts the data using AES and then uploads the ciphertext to the fog nodes. After receiving the ciphertext, the fog nodes generate k data blocks and x redundant data blocks using RS encoding. According to the storage capacity of fog nodes and cloud nodes, we design an allocation strategy for cloud-fog collaborative storage that stores some data in the fog nodes and uploads the rest of the data to the cloud nodes. This strategy is described in the next section.

3.3.2. Allocation Strategy for Cloud-Fog Collaborative Storage. As noted in the previous section, after receiving the ciphertext, the fog nodes generate k data blocks and x redundant data blocks by RS encoding. Then, we allocate $k + x$ data blocks according to the storage capacity of the fog nodes and the cloud nodes. Let $N_f = \{N_{f1}, N_{f2}, \ldots, N_{fn}\}$ represent fog nodes and $S_f = \{S_{f1}, S_{f2}, \ldots, S_{fn}\}$ represent the storage capacity of fog nodes. Similarly, let $Nc = \{N_{c1}, N_{c2}, \ldots, N_{cn}\}$ represent cloud nodes and $S_c = \{S_{c1}, S_{c2}, \ldots, S_{cn}\}$ represent the storage capacity of cloud nodes. In addition, because the storage capacity of the cloud nodes is greater than that of the fog nodes, we store most of the data in the cloud according to the storage capacity of the cloud nodes. To prevent data leakage, we store $x + 1$ block data blocks in fog nodes and store $k - 1$ data blocks in cloud nodes. The total time T_{total} required for cloud-fog collaborative storage is expressed as

$$T_{\text{total}} = T_{\text{cloud}} + T_{\text{fog}}, \tag{8}$$

where T_{cloud} is the storage time required for the cloud and T_{fog}, that for the fog.

The storage time required for each cloud node is expressed as

$$T_{ck} = t_{ck} \cdot (k-1) \cdot \frac{S_{ck}}{\sum_{j=1}^{n} S_{cj}}, \tag{9}$$

where T_{ck} is the storage time required for cloud node N_{ck} and $t_{ck}$is the storage time required when cloud node N_{ck} stores each data block.

Therefore, the storage time T_{cloud} required for the cloud is expressed as

$$T_{\text{cloud}} = \max\{T_{c1}, T_{c2}, \ldots, T_{cn}\}. \tag{10}$$

In addition, we analyze the two storage methods for the fog nodes as follows:

(1) The data is first stored in the first fog node; the remaining data is stored in the next fog node when the amount of data in the first node reaches an acceptable maximum, and so on.

At this time, the storage time T_{fog} required for the fog is expressed as

$$T_{\text{fog}} = \begin{cases} t_{f1} \cdot (x+1), & 0 < x+1 \le F_{1\_\max}, \\ t_{f1} \cdot F_{1\_\max} + t_{f2} \cdot (x+1-F_{1\_\max}), & F_{1\_\max} < x+1 \le F_{1\_\max} + F_{2\_\max}, \\ \dots & \\ t_{fn} \cdot \left(x+1-\sum_{i=1}^{n-1} F_{i\_\max}\right) + \sum_{i=1}^{n-1} t_{fi} \cdot F_{i\_\max}, & \sum_{i=1}^{n-1} F_{i\_\max} < x+1 \le \sum_{i=1}^{n} F_{i\_\max}, \end{cases} \tag{11}$$

where t_{fi} is the storage time required when fog node N_{fi} stores each data block and $F_{i\_\max}$ is the maximum storage required for fog node N_{fi}.

(2) We store data based on the storage capacity of the fog nodes. At this time, the storage time required for each fog node is expressed as

$$T_{fk} = t_{fk} \cdot (x+1) \cdot \frac{S_{fk}}{\sum_{i=1}^{n} S_{fi}}. \tag{12}$$

Therefore, the storage time T_{fog} required for the fog is expressed as

$$T_{\text{fog}} = \max\left\{T_{f1}, T_{f2}, \dots, T_{fn}\right\}. \tag{13}$$

After calculating T_{fog}, we can calculate the total time required for cloud-fog collaborative storage by substituting the results of T_{cloud} and T_{fog} into equation (8).

3.3.3. Security Analysis. For non-real-time processing of industrial data, we designed a data protection scheme for cloud-fog collaborative storage based on AES encryption and RS encoding. Some encoded data were stored in the fog nodes and the rest, in the cloud nodes, to realize multilayer data protection. This solution solves the problem of data leakage to a large extent. Here, we discuss the worst case. If the attackers are intelligent enough to steal k (or more than k) data blocks from the cloud and fog, they have to crack the encoding matrix in RS encoding and the AES algorithm to obtain the original data. However, it is very difficult for attackers to crack both the AES algorithm and the encoding matrix. Here, we consider the AES-128 algorithm as an example. It takes 2^{127} ns to crack the AES algorithm when decryption is performed once per nanosecond; this is impossible for attackers. Table 1 shows the degree of difficulty if attackers want to crack the encoding matrix.

Table 1 shows that it is difficult for attackers to crack the encoding matrix. In an industrial scenario, the numbers of source and redundant data blocks are very large; therefore, it is impossible for attackers to crack the encoding matrix in theory.

3.4. Efficiency Analysis. This section mainly analyzes the encoding efficiency and storage efficiency.

3.4.1. Encoding Efficiency. The RS algorithm performs the four fundamental arithmetic operations in the Galois field, in which the number of elements in the field must be greater than the sum of the number of source data blocks and redundant data blocks, that is, $2^w > n + m$, where n and m are the numbers of source data blocks and redundant data blocks, respectively. When the word size w is fixed, the time complexity of the RS algorithm with the Vandermonde matrix is $O(n^2)$; therefore, the encoding efficiency decreases with an increase in the number of the source data block. Generally, w is 8 or 16 because the computer stores bytes as 8 bits. Table 2 shows the data recovery performance when the number of data blocks n remains unchanged and w takes different values.

Table 2 shows that the data recovery performance is almost unaffected with an increase in w. Considering that a larger w value has higher encoding efficiency (processing more data blocks at a time) and that the number of source data blocks n is very large in the actual industrial scenario, we can try to choose the highest w allowed by the system.

3.4.2. Storage Efficiency. The storage efficiency is an important index of storage-related algorithms. A system with high storage efficiency can save the storage capacity as much as possible. According to the Storage Industry Networking Association's definition of storage efficiency, the storage efficiency E_{st} in this study is expressed as

$$E_{st} = \frac{n}{n+m}. \tag{14}$$

This equation shows that when n is larger and m is smaller, the storage efficiency becomes increasingly higher. Considering the large number of source data blocks n in the industrial scenario, a small number of redundant data blocks are needed to improve storage efficiency. In this study, for real-time industrial data, the minimum number of redundant data blocks can be selected as described in Section 3.2.4. For non-real-time industrial data, the minimum number of redundant data blocks can be selected according to the storage capacity of cloud nodes and fog nodes.

4. Results and Discussion

4.1. Experiments on Protection of Real-Time Industrial Data. This section verifies the performance of the proposed data protection scheme through simulation experiments.

4.1.1. Experimental Configuration. In the simulation experiment, we use the following 10 groups of data that obey a

Table 1: Degree of difficulty for cracking encoding matrix.

Word size of Galois field	Number of source data blocks k	Number of redundant data blocks m	Cracking time
8	6	1	2^{48}
8	6	2	2^{96}
16	6	1	2^{96}
16	6	2	2^{192}

Table 2: Comparison of processing time for different word lengths.

Word size of Galois field	Number of source data blocks k	Time (ms)
8	20	19.2
16	20	19.2
8	200	147.1
16	200	147.2

Table 3: Environmental parameters (real-time data).

Item	Parameter value
Operating system	Linux
Programming language	Python
CPU	Intel Core i5 2.30 GHz
Memory	8 GB
Hard disk	1 TB

normal distribution as location data: 10^5, 2×10^5, ..., 10^6. The data attribute value set is {1, 2, 3, ..., 100}. And we use the turbofan engine aging dataset provided by the NASA Prognostics Data Repository [31] as nonsensitive data. Table 3 shows the environmental parameters.

4.1.2. Experimental Results.

(1) M-RAPPOR

(1) *Performance of M-RAPPOR.* Figure 7 shows a comparison between the computational cost of M-RAPPOR and RAPPOR. We tested 10 sets of data. This figure shows that the improved M-RAPPOR algorithm has a lower computational cost because only one disturbance occurs in the data sender, and lasso regression is not conducted in the data collector. In addition, the communication cost refers to the cost of data transmission from the data sender to the data collector. M-RAPPOR and RAPPOR have the same communication cost which is related to the length k of Bloom filter. Communication cost increases with the increase of k. Because the main goal of this paper is to solve the problem of secure storage of industrial sensitive data, we think the communication cost is acceptable.

(2) *Relationship between Privacy Budget and Level of Privacy Protection.* According to the definition of local differential privacy, the level of data privacy protection mainly depends on the setting of the privacy budget ε, and the ε value is directly related to f_1, f_2. Figure 8 shows the relationship between f_1, f_2, ε, and the level of privacy protection. The value of the privacy budget ε is varied as 0.4, 2, and 4.6. The height of the blue column represents the true frequency distribution and that of the orange column, the statistical frequency distribution after M-RAPPOR.

This figure shows that with a smaller privacy budget such as $\varepsilon = 0.4$, the statistical results deviate from the real value to a large extent, whereas with a larger privacy budget such as $\varepsilon = 4.6$, the statistical results are closer to the real value.

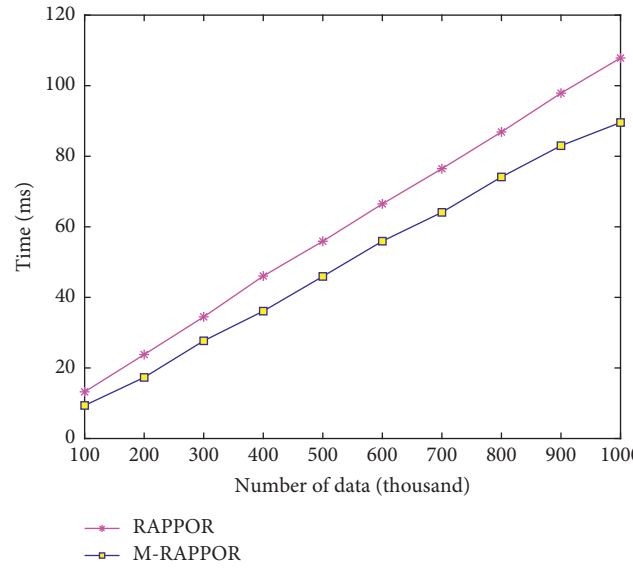

Figure 7: Computational cost of RAPPOR and M-RAPPOR.

Therefore, the privacy budget is negatively related to the level of privacy protection. The lower the privacy budget is, the lower the data availability is and the higher the level of privacy protection is. In the predictive maintenance scenario, we can appropriately select a higher privacy budget that not only ensures data availability but also protects the specific location information of the factory equipment.

(2) Reed–Solomon Encoding

RS encoding consists of the following steps.

(1) *Determination of Equalized Storage and Nonequalized Storage.* Figure 9 shows the relationships among the equalized storage, nonequalized storage, and decoding time. This graph shows that the decoding time of the data blocks stored in the equalized mode is slightly better than that of the blocks stored in the nonequalized mode. However, in an actual industrial scenario, not all storage equipment has the same storage capacity; therefore, the benefits of allocating data blocks according to the

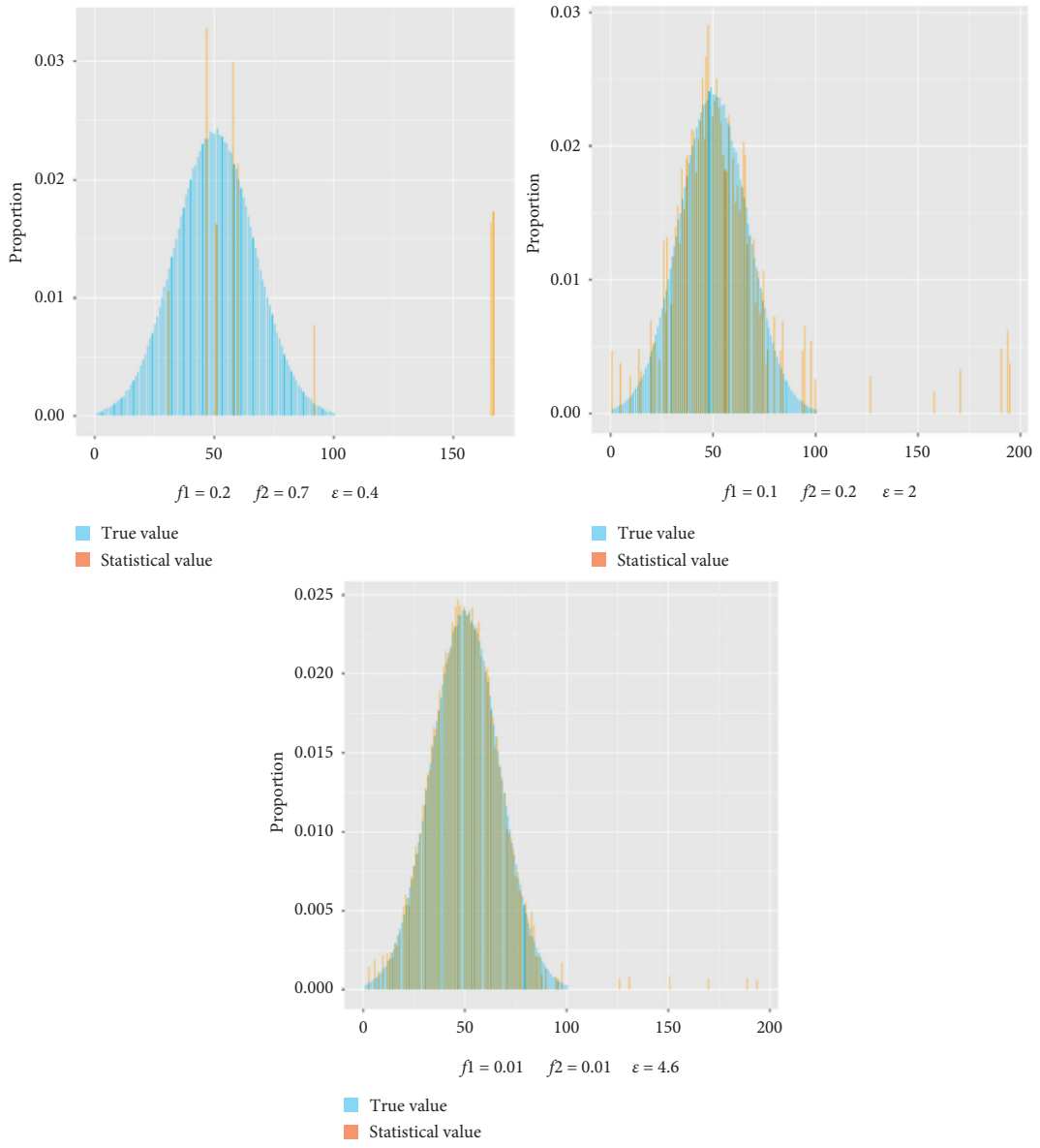

Figure 8: Frequency statistics results under different privacy budgets.

configuration of different pieces of equipment are greater than that of equalization. Therefore, we adopt nonequalized storage to optimize overall performance.

(2) *Selection of Optimal Redundant Data Blocks.* Figure 10 shows the relationship between the encoding time and the decoding time with a change in the number of redundant data blocks. We vary the number of redundant data blocks from 0 to 30. With an increase in the number of redundant data blocks, the encoding time increases whereas the decoding time basically remains unchanged. This is because the encoding time is directly related to the encoding matrix. The encoding matrix comprises a unit matrix and a Vandermonde matrix. When the number of redundant data blocks increases, the dimension of the Vandermonde matrix increases, leading to an increase in the operation cost and encoding time. In addition, because no data block has been removed in this test, it is not necessary to recalculate in the decoding process, data can be extracted directly from the equipment, and decoding time is basically unchanged. Therefore, we should select the minimum based on the data recoverability.

(3) *Verification of Robustness of Distributed Storage.* To reflect the advantages of distributed storage and robustness of the scheme, we have designed a local equipment failure scenario. Taking Figure 6 as an example, assume that there are six local devices ($n = 6$) and that the storage capacities are 1 MB, 1 MB, 4 MB, 6 MB, 8 MB, and 10 MB, respectively. The number of encoded real-time processing data blocks is 60 ($b = 60$). According to equation (2), the number of data blocks stored by each equipment type is 2, 2, 8, 12, 16, and 20, respectively. According

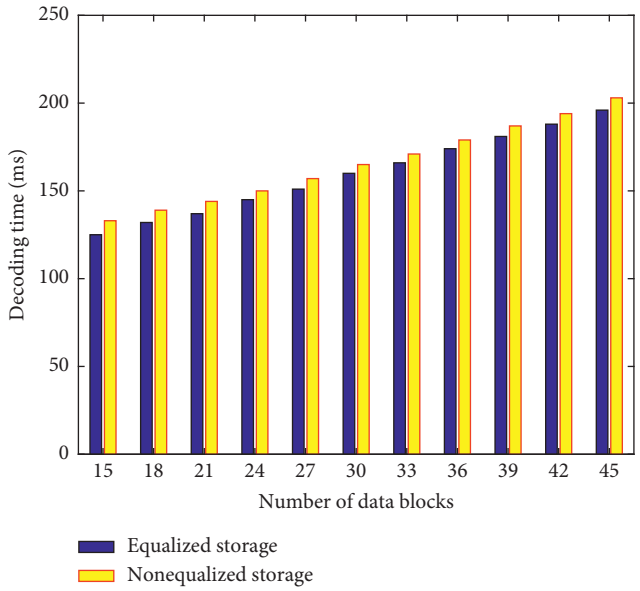

Figure 9: Relationship between equalized storage, nonequalized storage, and decoding time.

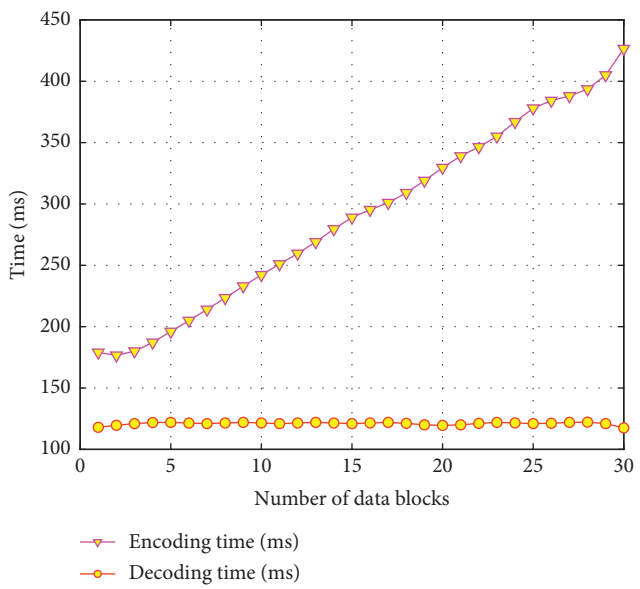

Figure 10: Relationship between encoding time and decoding time as the change of redundant data blocks.

to inequality (3) and equation (4), the optimal value of the number of redundant data blocks m is obtained as 20. In our solution, if any device fails, we can recover the complete data. To verify this, we performed tests six times. Each device suffers a device failure, and the test results are shown in Figure 11. This figure shows that if any one of the six local equipment types fails, we can recover the complete data. In addition, we can understand the data loss caused by equipment failure. As the amount of lost data increases, the decoding time increases. This is because the lost data need to be solved by the redundant data block combined with the corresponding row in the decoding matrix to obtain the complete equation. The larger the amount of data that is lost, the more difficult to solve the equations and the greater the amount of time consumed.

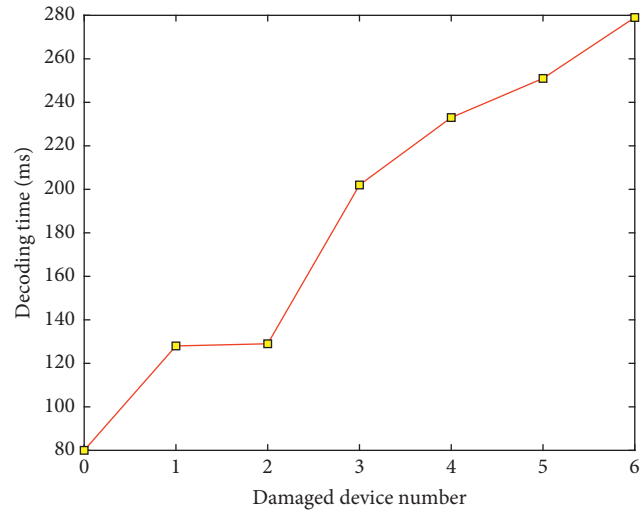

Figure 11: Relationship between different local equipment failures and decoding time.

Table 4: Environmental parameters (non-real-time data).

Item	Parameter value
Operating system	Windows 10
Programming language	Java
CPU	Intel Core i5 3.30 GHz
Memory	8 GB
Hard disk	1 TB

4.2. Experiments on Protection of Non-Real-Time Industrial Data

4.2.1. Experimental Configuration

(1) Dataset. We chose an industrial production dataset provided by Bosch [32] and selected 36 MB of non-real-time data from this dataset. We stored $x + 1$ data blocks in the fog nodes. This scheme can protect data and reduce the storage pressure on the fog nodes.

(2) Experimental Environment. Table 4 shows the environmental parameters. In addition, we used MATLAB (MathWorks, Natick, MA, US) as the data analysis tool.

4.2.2. Experimental Results. This section presents the results of our cloud-fog collaborative storage experiments and comparisons of the time required for cloud-fog collaborative storage when different storage methods are adopted for the fog nodes. After receiving the ciphertext, the fog nodes generate k data blocks and x redundant data blocks by RS encoding. Considering that the processing capacity of fog nodes is better than that of local equipment, we set x to 60.

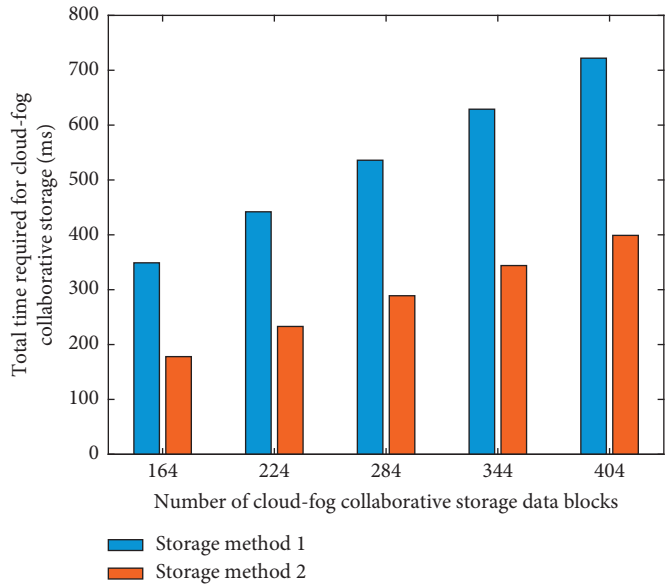

Figure 12: Total time required for cloud-fog collaborative storage using storage method 1 and storage method 2 for fog nodes.

Thus, we store $x + 1$ ($x + 1 = 61$) data blocks in the fog nodes and upload the remaining $k - 1$ data blocks to the cloud nodes. We assume that there are three cloud nodes N_{c1}, N_{c2}, N_{c3}, where each node has a storage capacity of 200 MB. The time required for the storage of each data block is 2 ms. There are three fog nodes N_{f1}, N_{f2}, N_{f3} having storage capacities of 30 MB, 30 MB, and 40 MB, respectively, and the maximum number of data blocks storing $F_{1\_\max}, F_{2\_\max}, F_{3\_\max}$ is 60, 60, and 70, respectively. The time t_{f1}, t_{f2}, t_{f3} required for the storage of each data block is 6 ms, 6 ms, and 5.5 ms, respectively. From equations (11) –(13), we can calculate the storage time required when the fog nodes adopt two different storage methods. According to equations (9) and (10), we can calculate the storage time required for the cloud nodes. Finally, according to equation (8), we can calculate the total time required for cloud-fog collaborative storage. Figure 12 shows the total time required for cloud-fog collaborative storage when the fog nodes adopt storage method 1 and storage method 2. This figure shows that the total time required for cloud-fog collaborative storage is less for storage method 2, and as the number of data blocks that the fog nodes need to encode increases, the difference in the total storage time required by the two different storage methods increases; that is, the advantage of storage method 2 is greater. Considering the large amount of non-real-time processing data in the industrial scene, we choose storage method 2 in the fog nodes to achieve the best performance for the entire cloud-fog collaborative storage solution.

5. Conclusions

At present, most industrial enterprises do not have perfect industrial data protection measures and lack a complete and effective industrial data protection solution. However, relying only on fog computing or cloud computing cannot fully protect industrial data. We proposed a three-layer protection framework with local/fog/cloud storage for sensitive industrial data and defined a corresponding threat model. For real-time sensitive industrial data, we designed a data protection scheme based on the improved local differential privacy algorithm M-RAPPOR and RS encoding. We desensitized and encoded the data in local equipment. We then adopted the optimal scheme for distributed storage in local equipment to realize low cost, high efficiency, and intelligent data protection. For non-real-time sensitive industrial data, we designed a data protection scheme for cloud-fog collaborative storage based on AES encryption and RS encoding. Some encoded data were stored in the fog nodes and the rest, in the cloud nodes, to realize multilayer data protection. The feasibility of our scheme has been validated through security analysis and experimental evaluations.

Data Availability

Reference [32] refers to the dataset used in this research. The dataset also can be accessed from the web page https://www.kaggle.com/c/bosch-production-line-performance/data.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was partly supported by the National Natural Science Foundation of China (61872015), Qinghai Province Natural Science Foundation (2017-ZJ-91), Beijing Natural Science Foundation-Haidian Original Innovation Joint Fund (19L2020), Foundation of Science and Technology on Information Assurance Laboratory (614211204031117), Industrial Internet Innovation and Development Project

(Typical Application and Promotion Project of the Security Technology for the Electronics Industry) of the Ministry of Industry and Information Technology of China in 2018, Foundation of Shanxi Key Laboratory of Network and System Security (NSSOF1900105), and International Research Cooperation Seed Fund of Beijing University of Technology (2018-B9).

References

[1] M. Tseng, T. Edmunds, L. Canaran et al., *Introduction to Edge Computing in IIoT*, Industrial Internet Consortium, Needham, MA, USA, 2018.

[2] H. Hui, C. Zhou, S. Xu, and F. Lin, "A novel secure data transmission scheme in industrial internet of things," *China Communications*, vol. 17, no. 1, pp. 73–88, 2020.

[3] B. Wang, "Facebook has another data breach scandal and is questioned that it cannot fully protect user information," 2019, http://www.qlmoney.com/content/20190404-347582.html.

[4] K.-S. Wong and M. H. Kim, "Privacy protection for data-driven smart manufacturing systems," *International Journal of Web Services Research*, vol. 14, no. 3, pp. 17–32, 2017.

[5] C. Dwork and J. Lei, "Differential privacy and robust statistics," in *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, pp. 371–380, Bethesda, MA, USA, May–June 2009.

[6] K. Kenthapadi, A. Korolova, I. Mironov, and N. Mishra, "Privacy via the johnson-lindenstrauss transform," *Privacy Confidentiality*, vol. 5, no. 1, pp. 39–71, 2013.

[7] N. Mohammed, D. Alhadidi, B. C. M. Fung, and M. Debbabi, "Secure two-party differentially private data release for vertically partitioned data," *IEEE Transactions on Dependable and Secure Computing*, vol. 11, no. 1, pp. 59–71, 2014.

[8] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, "Local privacy and statistical minimax rates," in *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 429–438, Berkeley, CA, USA, October 2013.

[9] U. Erlingsson, "RAPPOR: Randomized aggregatable privacy preserving ordinal response," in *Proceedings of the 21st ACM Conference on Computer and Communications Security*, pp. 1054–1067, Chicago, IL, USA, October 2014.

[10] B. Gu, V. S. Sheng, Z. Wang et al., "Incremental learning for *v*-support vector regression," *Neural Networks*, vol. 67, pp. 140–150, 2015.

[11] C. Xu, J. Ren, D. Zhang, and Y. Zhang, "Distilling at the edge: a local differential privacy obfuscation framework for IoT data analytics," *IEEE Communications Magazine*, vol. 56, no. 8, pp. 20–25, 2018.

[12] T. Wang, J. Zhou, X. Chen, G. Wang, A. Liu, and Y. Liu, "A three-layer privacy preserving cloud storage scheme based on computational intelligence in fog computing," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 3–12, 2018.

[13] M. Du, K. Wang, X. Liu, S. Guo, and Y. Zhang, "A differential privacy-based query model for sustainable fog data centers," *IEEE Transactions on Sustainable Computing*, vol. 4, no. 2, pp. 145–155, 2019.

[14] L. Lyu, K. Nandakumar, B. Rubinstein, J. Jin, J. Bedo, and M. Palaniswami, "PPFA: privacy preserving fog-enabled aggregation in smart grid," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3733–3744, 2018.

[15] Q. Huang, Y. Yang, and L. Wang, "Secure data access control with ciphertext update and computation outsourcing in fog computing for Internet of Things," *IEEE Access*, vol. 5, no. 99, pp. 12941–12950, 2017.

[16] R. Lu, K. Heung, A. H. Lashkari, and A. A. Ghorbani, "A lightweight privacy-preserving data aggregation scheme for fog computing-enhanced IoT," *IEEE Access*, vol. 5, pp. 3302–3312, 2017.

[17] T. Ghorbani, J. Zeng, M. Z. A. Bhuiyan et al., "Trajectory privacy preservation based on a fog structure for cloud location services," *IEEE Access*, vol. 5, pp. 7692–7701, 2017.

[18] S. Kulkarni, S. Saha, and R. Hockenbury, "Preserving privacy in sensor-fog networks," in *Proceedings of The 9th International Conference for Internet Technology and Secured Transactions (ICITST)*, vol. 96–99, London, UK, December 2014.

[19] Q. Hou, Y. Wu, W. Zheng, and G. Yang, "A method on protection of user data privacy in cloud storage platform," *Journal of Computer Research and Development*, vol. 48, no. 7, pp. 1146–1154, 2011.

[20] G. Feng, "A data privacy protection scheme of cloud storage," *Software Guide*, vol. 14, no. 12, pp. 174–176, 2015.

[21] Z. Fu, X. Wu, C. Guan, X. Sun, and K. Ren, "Toward efficient multi-keyword fuzzy search over encrypted outsourced data with accuracy improvement," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 12, pp. 2706–2716, 2016.

[22] Z. Fu, K. Ren, J. Shu, X. Sun, and F. Huang, "Enabling personalized search over encrypted outsourced data with efficiency improvement," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 9, pp. 2546–2559, 2016.

[23] Z. Fu, F. Huang, X. Sun, A. Vasilakos, and C.-N. Yang, "Enabling semantic search based on conceptual graphs over encrypted out-sourced data," *IEEE Transactions on Services Computing*, vol. 12, no. 5, pp. 813–823, 2016c.

[24] Z. Xia, X. Wang, X. Sun, and Q. Wang, "A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 2, pp. 340–352, 2016.

[25] G. Kulkarni, R. Waghmare, R. Palwe et al., "Cloud storage architecture," in *Proceedings of the 2012 7th International Conference on Telecommunication Systems, Services, and Applications*, pp. 76–81, Bali, Indonesia, October 2012.

[26] C. Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for secure cloud storage," *IEEE Transactions on Computers*, vol. 62, no. 2, pp. 362–375, 2013.

[27] J. Shen, J. Shen, X. Chen, X. Huang, and W. Susilo, "An efficient public auditing protocol with novel dynamic structure for cloud data," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 10, pp. 2402–2415, 2017.

[28] X. Q. Pham and E. N. Huh, "Towards task scheduling in a cloud-fog computing system," in *Proceedings of the 2016 18th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pp. 1–4, Kanazawa, Japan, October 2016.

[29] R. Deng, R. Lu, C. Lai et al., "Towards power consumption-delay tradeoff by workload allocation in cloud-fog computing," in *Proceedings of the 2015 IEEE International Conference on Communications (ICC)*, pp. 3909–3914, London, UK, June 2015.

[30] K. Bierzynski, A. Escobar, and M. Eberl, "Cloud, fog and edge: cooperation for the future?," in *Proceedings of the 2017 Second IEEE International Conference on Fog and Mobile Edge Computing (FMEC)*, pp. 62–67, Valencia, Spain, May 2017.

[31] V. Mathew, T. Toby, V. Singh et al., "Prediction of remaining useful lifetime (RUL) of turbofan engine using machine learning," in *Proceedings of the 2017 IEEE International Conference on Circuits and Systems (ICCS)*, pp. 306–311, Thiruvananthapuram, India, December 2017.

[32] A. Mangal and N. Kumar, "Using big data to enhance the Bosch production line performance: a kaggle challenge," in *Proceedings of the 2016 IEEE Big Data Conference*, pp. 2029–2035, Washington, DC, USA, December 2016.