Computing Time Series Data During Index Based De-Duplication of Industrial IoT Data in Cloud Environment

Muthunagai S U (muthunagaisu@gmail.com)
Sri Venkateswara College of Engineering https://orcid.org/0000-0002-9171-8784
Anitha R
Sri Venkateswara College of Engineering

Research Article

Keywords: Industrial IoT, De-duplication, Merkle tree, Reckoning of occurrences, Time series analysis.

Posted Date: December 8th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-850002/v1

Computing Time Series data during Index Based De-duplication of Industrial IoT data in Cloud Environment

Muthunagai S U[1*] Anitha R[2]

[1]Assistant Professor, Department of CSE,
Sri Venkateswara College of Engineering, Sriperumbudur, India
muthunagai@svce.ac.in

[2]Professor, Department of CSE,
Sri Venkateswara College of Engineering, Sriperumbudur, India
ranitha@svce.ac.in

*Corresponding Author: Muthunagai S U (muthunagai@svce.ac.in)

Abstract

As a result of the development in Industry 4.0, the data generated within the Industries are increasing rapidly every day to attain the innovative environment within the industry through maximal asset utilization. Meanwhile, the redundancy rate in the server is also increasing, which has an impact on the storage as well as in the analysis of data. Most existing de-duplication techniques partition the data with respect to memory. However if the time period is considered for partition, time-series analysis would be achieved during the de-duplication process. To address the above issue, the proposed work presents the Index Based De-duplication technique with Categorized Region Method for computing time-series data. The Merkle Tree with a super feature called reckoning of occurrence is combined in the proposed system to rapidly identify the existence of similar data in the distributed system with an accurate existence count that significantly helps in predicting the future drifts of the industrial environment. Finally, the proposed system also concludes with optimal transportation cost to reach the storage nodes in the cloud using MODI method. The experimental results reveal that the proposed model is efficient since it facilitates less memory and less computation overhead. The proposed technique achieves space reduction by 98%, reduces the computation overhead during analysis by 55%, and increases the efficacy of cloud storage by 60%.

Keywords: Industrial IoT, De-duplication, Merkle tree, Reckoning of occurrences, Time series analysis.

1. Introduction

Industrial IoT (IIoT) is a framework that outlines various physical components, operational groups and procedures, network alignment and the data layouts to be used. In nature, the Internet of Things (IoT) framework is the arrangement of various components, which include receptors, regulations, initiators, virtual assistance, and zones. IIoT has been developed as a significant arena for research by resonating IoT data associated problems to be resolved especially in cloud storage environments. Numerous conflicts raised due to IoT based data accumulation in cloud storage are non-centralized execution with the combined administration of framework reserves, multi-dweller repository with solo functionality, and expandability with flexibility. Many issues raised due to huge and unstructured data dispensation at various stages like data depiction, data accumulation and analysis [1].The IIoT applications like robotics, supply chain, monitoring and Industries related to automotive, manufacturing, retail, aerospace are currently evolving IIoT with advanced features. According to the Meticulous Research organization, the IIoT market is expected to rise at a Component Annual Growth Rate (CAGR) of 16.7% during 2022-2027. Which is also estimated to attain $263.4 billion by 2027 [2]. The IoT sensors embedded in the various industrial environments generate monitored data sequentially upon the prompt of extraneous action. On the other hand, data produced by several sensor nodes must to be grouped, accumulated, examined and envisioned to attain The IoT sensor network's model move towards emergent technologies like edge, fog and cloud computing which undergoes high complexity in data dispensation, data fusion and sensor data analytics [3]. One of the most challenging problems in the IIoT environment is configuring the fog system in optimum way as industrial strategies are varied by means of provision and demand [4].

In manufacturing industries, a massive amount of redundant information is delivered to the information appraisal zone. The allocation of abundant space for this redundant data is not sufficient. It augments the repository capability and reduces the efficacy of the manufacturing system. So, the data de-duplication schematic is needed where the unused or same information is obtained and removed. Besides, intelligent decision-making models contribute to handling things such as assistants by means of a pattern-based decision-making system which is recognized by providing and incorporating the whole firm data [5]. Industries need time series data to predict the scenarios held up in the industrial environment. However during the de-duplication process redundant data will be removed, if the amount of redundant data is measured prior to elimination with respect to timestamp time series data can be computed. Accumulating the core facts in IIoT equipment regionally is not advisable when the resultant equipment power is concerned. This also involves the strict restriction of the garage area. Therefore, the equipment is not to be relied upon and prone to a large number of dangers as a result of framework provision in distant and neglected areas. Hence, the observed industrial data is stored in a cloud for high scalability and flexibility. To store massive amount of IIoT data in cloud, deduplication technique with optimal path is required.

To overcome the limitations of existing de-duplication schemes towards IIoT, the proposed system creates the newest approach to achieve de-duplication with periodic monitored data. The contribution of the research work are as follows

- The partitioning of data is carried out based on time interval to produce time series data which reduces computation overhead during the decision making process.

- A secure index based de-duplication system with reckoning of occurrence is proposed. It reduces more space occupied by redundant by storing a unique value of monitored sensor data with existence count.
- An optimal transportation cost to store industrial monitored data is determined using Modified Distribution Method to improve the efficacy of cloud storage

The paper is compiled in such a way that section 1 defines the latest parameters in IIoT. The subsequent section 2 outlines the related research work which has been carried out on IIoT and existing de-duplication scheme. The section 3 consists of the framework and the proposed architecture. The experimental set-up and performance evaluation for the proposed system is determined in section 4 and 5. Lastly, at section 6 the conclusion of the paper is provided.

2. Related Work

Table 1. Summary of Related work

Authors	Proposed Scheme	Method	Limitations
Zheng et. al [10]	Certificateless proxy re-encryption based data de-duplication scheme	Improves the detection mechanism at decryption end to exactly identify the redundant files	Decrease in network lifetime
Xia et. al [11]	Pipelined and Parallelized data de-duplication scheme	During the de-duplication practice the chunks were created through a pipelined process.	Less time consuming but elongated processes were carried out to chunk the data.
Fu et. al [12]	Scalable inline distributed de-duplication scheme	This scheme facilitates intra-node de-duplication with two-tiered routing decision by developing application awareness, data similarity and range.	Increase in computation overhead
Xia et. al [13]	Duplicate-Adjacency based Resemblance Detection	The enhanced resemblance detection reduces redundancy rate with less overhead	Major problems have been raised during the segmentation of data.
Yan et. al [14]	De-duplication scheme on encrypted data	Proxy re-encryption is represented to carry out the de-duplication and data deletion process.	High security for big data with high computation cost.
Yu et. al [15]	Privacy aware de-duplication protocols	The zero-knowledge de-duplication framework prevents attackers from healing the information gaining the existence status.	Two de-duplication protocols achieve two- sided privacy with increase in communication overhead.
Ni et. al [16]	Fog-assisted mobile crowdsensing framework	This framework empowers the fog nodes in the IoT environment to sense and remove duplicate data through a de-duplication system embedded in it.	It provides resistance across brute-force and duplicate-replay attacks with less de-duplication ratio.
Tian et. al [17]	Randomized de-duplication scheme	The client-side de-duplication system guarantee the confidentiality of outsourced data	Even if it prevents collusive authentication attacks and offline attacks raised by the eternal hackers it fails to achieve optimality in the data storage arena.
Jiang et. al [18]	Secure data de-duplication scheme	It provisions both file level and block level data de-duplication	This scheme achieves consistency and provides mutual agreement with dynamic ownership management
Gao et. al [19]	De-duplication scheme based on threshold dynamic adjustment	This scheme determines the sensitivity of various data with privacy score during uploading of data into the storage system	This scheme supports de-duplication with lower privacy and not for higher extent.
Sharma et. al [20]	Secure de-duplication scheme for fog assisted nodes.	ECC based hybrid multiplier and Multi-Objective based whale Optimization algorithm ensures clustering of fog nodes.	Achieves de-duplication for IIoT data with less computation overhead but the decision making process with deduplicate data is quite complex.
Fu et. al [21]	Secure data storage and retrieval Scheme	RF Tree , AVL tree considered for efficient storage and indexing	It helps in optimizing the storage space and efficient searching mechanism but time complexity is more in decision making.
Ellapan et. al [22]	Dynamic Chunking algorithm	Window size can be adjusted with dynamic prime algorithm for de-duplication process	Memory is the only factor considered during chunking the data.

Reducing storage space allocated for industrial data is one of the main challenges in this era. If any intelligent actions were performed concurrently during the elimination of redundant data will be more advantageous to the industrial environment. Inspired to move in this direction, this research work proposes an index based de-duplication system with reckoning of occurrence feature to produce time series data through that the industrial environment can be profited. This section discusses the evolution of IIoT from WSN to fog with the importance of de-duplication scheme and also summarizes the various approaches involved in the de-duplication system. The IoT has produced a great impact in the modern world due to the high contribution of Wireless Sensor Network (WSN). Fog computing replaces computation carried out by WSN as there is

confined energy usage at sensor nodes. Fog brings storage resources and computation nearer to users. The evolution of Industry 4.0 has been facilitated due to adoption of Cyber Physical System (CSP), IoT, cloud and Artificial Intelligence. Furthermore, many industrial applications require a decision making model [6], [7]. Internet users have been raised dramatically in recent years due to provisioning of the Internet all over the world. Many people were invading online through many applications. Hanging with social media becomes part of their daily life. Besides, due to this COVID pandemic, many jobs, educational systems ere captivating the advantages of technology to complete their work online even during these tough times. The tremendous growth in the usage of mobile and web applications leads to exponential increase in the data across the globe. Hence, storage space is essential for storing the data has become the most important concern [8]. Provisioning of storage optimization techniques became a vital constraint to huge storage capacities like cloud storage. De-duplication is a storage optimization technique which evades accumulating identical replicas of data [9]. The main task to be performed in de-duplication is partitioning of data [11], [18], [22]. As de-duplication is carried out with protective data various security measures are considered [10], [12-15], [19]. While decrypting the data for the de-duplication process many attackers may intrude to steal the data which is addressed with various potentials in [16], [17]. The secure storage mechanism for IIoT (SDSSIIoT) and de-duplication carried with $2FBO^2$ for IIoT data (FaCIIoT) is stated in [21], [20]. Many existing de-duplication work considers memory and security as the prime factor which is depicted in **Table.1**. However, with this, the decision making process cannot be achieved rapidly. To address these challenges, the proposed system implements index based de-duplication with a super feature called reckoning of occurrence reckoning for each time interval.

3. Index Based De-duplication Using Merkle Tree with Reckoning of Occurrence

The architecture diagram for the proposed model for the Industrial IoT cloud environment is shown in **Fig.1** and discussed in detail in this section. The proposed scheme for the Industry 4.0 consists of three components which are Partitioning, De-duplication, and Optimal path determination to store the data in a cloud environment. In partitioning, the proposed system employs the Categorized Region Method, which forms the regions in the extracted sensor values prior to the execution of de-duplication. In De-duplication, the redundant data present in the IIoT data is reduced using index-based de-dup the three components and the design applications are mentioned.

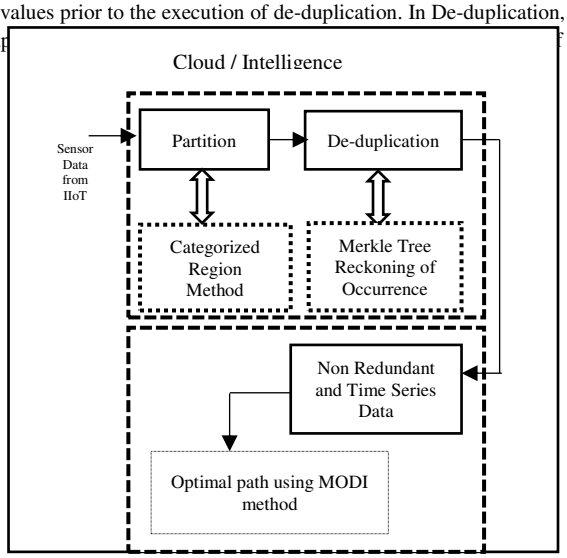

Fig.1. Framework of proposed de-duplication system

3.1 Categorizing Region Method

Due to the emergence of the Industry 4.0, various processes are enhanced, resulting in enabling the data analysts to address the problems such as data analysis and predictive maintenance. The performance of predictive analytics with the non-redundant sensor data is an essential thing as the data is collected and organized periodically. For example, In a gas industry, if any gas leakage in the industry occurs the industrial environment may get affected but none can predict accurately how worst it affects humans as well as the industrial environment in that particular period. Hence, this Categorized Region Method (CRM) fragments the collected sensor values into several parts with the time interval and boundary value (bv) using the chi-square test. Firstly, the data collected is partitioned with time period which are regions and in each region the sub regions were formed with boundary value and chi-square test which is presented in Algorithm.1. Using this chi-square test, the chunks are made in the sensor values, and the values in each area are sorted prior to the formation of the region. The variation between the initial value and the last value of i^{th} region (r_i) depends upon the presence of redundant values in the region. When the existence of the redundant value is high in the data collected, then the value in the regions is defined with transitive closure i.e. $r_i(v(f)) \approx r_i(v(l))$. When the records are sorted using the attributes containing the redundant values, the duplicate records tend to be located in the neighbor records. When the last value in the i^{th} region exists in $i+1^{th}$ region, then the region i is enhanced with the inclusion of similar values that exist in $i+1^{th}$ region. Then, the following criteria $s(r_{i+1}) < s(r_i)$ is achieved and notation are listed in **Table. 2**. Where the size of the regions is modified depending on the accumulation of values in the

region. In the categorized region phase, the time attribute is used for generating the regions. When the size of the data collected in a particular time period is large, the regions are subdivided with the boundary value. The boundary value (bv) is calculated by,

$$bv = \left\lfloor \frac{(\mu - s_1)}{\frac{\sigma}{\sqrt{n}}} \right\rfloor * t \tag{1}$$

$s_1 = \frac{var-SD}{n}$, $t = 2 * (l(r)/\sqrt[3]{n})$

The regions are categorized into subregions with the help of bv . In each region, the critical value is calculated. As a result, the irrelevant and noisy data is removed from the observed values to ensure accurate results. The chi-square test determines the region in the proposed model as χ^2 provides a measure of deviation between observed and expected value. The number of observed elements o_i in the categorized region cr_i for i=1, 2, 3 …y is encountered. Based on the null hypothesis, the expected number of elements E_i to be found in the categorized region cr_i for i=1, 2, 3 …y is calculated. The null hypothesis is a default hypothesis that measures the quantity to be zero or null. Where the quantity is different in two situations, the chi-square statistic is given as

$$\chi^2 = \sum_{i=1}^{y} \frac{(O(r_i) - E(r_i))^2}{E(r_i)} \tag{2}$$

The expected value $E(r_i)$ required for chi-square calculation is determined using $Max(O_i)$ and $Min(O_i)$ maximum value, which lies in the observed data. $E(r_i)$ value lies between $Min(O_i)$ and $Max(O_i)$. This chi-square test yields excellent results with large database n, and hence it is selected for the IIoT data.

Table.2 List of Notations

Notation	Description
bv	*Boundary value*
r_i	*i^{th} region*
$l(r)$	*Length of records*
$r_i(v(f))$	*First record value in the i^{th} region*
$r_i(v(l))$	*Last record value in the i^{th} region*
$s(r_i)$	*Size of region*
χ^2	*Chi-square test value*
o_i	*Observed values*
$O(r_i)$	*Observed values at the i^{th} region*
E_i	*Expected value*
$E(r_i)$	*Expected value at the i^{th} region*
cr_i	*Categorized region*
$Max(O_i)$	*Maximum Value in the observed values*
$Min(O_i)$	*Minimum Value in the observed values*

The CRM practices a succession of regions to derive near the location of the border values aggressively. At the same time, the approach uses different size ambient procedures in 2 stages: (1) an enhancement stage to procure a number of capable replicas along with a minimum number of inexpensive span computations and (2) a retrenchment stage to determine the border values inside the conclusive size from the initial stage.

Algorithm 1 Partitioning of regions using CRM

Input: *Observed sensor value from Industrial IoT*
Output: *Partition of regions with a time Interval*

Begin

1. *Initialize boundary value, sub_region, distance*
2. *Partition the region with the time period*
3. *Each region is again divided into k sub_regions with boundary value*
4. *Perform chi-square test under each sub-region*
5. *Compute the distance between first and last records in each sub_region*
6. *If the last value of sub_region n and the initial value of sub_region n+1 is similar,*
   then
   - ***a.*** *Enhance the sub_region n with inclusion of similar values from sub_region n+1*
   - *b. Retrench the sub_region n+1 with no. of values passed to sub_region n*

   else
   - *a. Repeat the step from 4 to 7 until all the regions are sorted.*
7. *End*

The algorithm for partitioning the data in a regional-wise is done using CRM method. The resultant regions from the CRM is passed to the Index based de-duplication to realize the main goal of the proposed work.

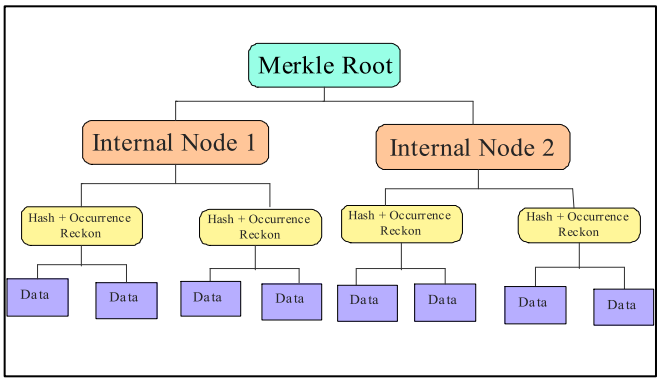

Fig. 2 Representation of Merkle Tree with Occurrence Reckoning

3.2 Index based data De-duplication for IIoT

This is a methodology to eliminate redundant copies of data existing within the database. It is also termed as single-instance data storage that improves the efficient utilization of memory and CPU. It also reduces the transmission bytes during the transfer of data in the network. In most cases, the sensor node generates a large number of redundant values. Such redundant data needs to be eliminated. In a traditional de-duplication system, the author provides various techniques to identify and remove the redundant data, but none has proposed a scheme to count the duplicate existence through which time series data can be produced. With the count of duplicate existences in each time period, the activities in industries can be predicted as like time-series analysis. On considering the above issue, the proposed system provides the time series analytical data for industrial environment while de-duplication is carried for the observed sensor values in a cloud storage. To perform the proposed work, the system uses Merkle Tree with the reckoning of occurrence feature in it. Once the regions and sub regions are partitioned based on the time period and boundary value, the observed sensor values in the region undergoes indexed-based de-duplication. The indexed-based de-duplication is carried out by Merkle Tree, which is used to store unique values with dynamic indexing on multiple levels. It generates a hash value for each data that resides in the leaf node and stores them in the non-leaf node called the internal node. Along with the hash value in the internal node, the proposed work employs the feature called reckoning of occurrence to measure the multiple existences of an each observed value in the Industrial environment. Due to this additional feature, the proposed de-duplication contains unique values with a count of number of times the data hits the tree structure for leaf node appearance that leads to effective indexing.

Merkle Hash Tree is a tree data structure containing hash-based values. It also simplifies the hash inventory. It is a non-linear data structure i.e. tree data structure where every leaf point holds data. Every non-leaf point contains hashes of its children, i.e., non-leaf nodes store hashes of data. The non-leaf nodes are represented as l_s and r_s . Here, the $l_s$left subtree contains left child nodes and r_s is the right subtree contains right child nodes. The non-leaf nodes in the tree are allocated to store the hash value of each data. The leaf nodes of both left and right nodes in the subtrees range from 1 to n-1. For instance, the origin of the tree i.e. Merkle root consists of n leaves then its key varies from 0 to $n-1$. Merkle Tree can be mathematically represented as.

$$\text{MT= RT } (In(l_s) * L(r_s)) \tag{3}$$

The Proposed de-duplication technique is constructed as follows

1) Generation of Internal node and leaf node

The bottom-most layer of the Merkle tree contains a leaf node with data. The numerous data collected from the Industrial environment is accumulated in the leaf node of the Merkle tree. When the leaf nodule is identified, a hash value is generated and stored in the internal node, which acts as a layer above the leaf node. SHA-3 512 is used in the hash value generation. The leaf node containing data is secured with 512 addressing mode during the hash value generation. Every observed value is allotted with 512-bit hash value and the recurrence of data is verified with the hash value. Correspondingly, the hashing advances at each stage, resulting in the attainment of elevated stages till the origin node called the 'Merkle root' is reached. As a result of the connection of hashes resembling a tree, the system comprises the data of all the operation hashes which are found in the node. It produces a single-level hash parameter which facilitates justifying all the parameters existing in the node.

2) Reckoning of Occurrence

To predict the future inclinations of the industrial environment as like time series analysis, the occurrence of each value need to counted with periodic intervals. With this maxim, the reckoning feature is employed along with Merkle Tree as a part of the proposed work, which is presented in **Fig. 2.** During the generation of the hash value for the leaf node, the existence factor for the leaf node is initialized as one for the first incidence. When the same data hits the tree for space, the hash value generated matches with the existing leaf node, results in the increment of existence count

before the de-duplication process is carried out. When several happenings of the same value persist in a particular time period, the Merkle tree considers only one instance, and incremented reckoned occurrence value is stored in the internal node along with the hash value of the leaf node.

3) Updation of Path between leaf node and root

Subsequent to the penetration of the value, a leaf node, the updation of the path, and the number of incidences of each observed value need to be reorganized periodically that is explained in Algorithm2. Thus the proposed system consistently maintains the path from Leaf nodes l_s and r_s to root node along with the occurrence factor. Since the construction of Merkle tree begins from the bottom-most layer of leaf nodes, the duplicate existence is also updated with a hash value of data as a separate field in the internal nodes. When indexing begins from the root node, along with the path to reach the data node location, the several existences of such data during the time period also be provided. Due to several hit to the tree by several times, the path established to find the nodes is accessed frequently which helps in improving the search mechanism.

Algorithm 2 Deduplication using Merkle Tree with Reckoning of Occurrence

Input: *Observed sensor value from Industrial IoT*
Output: *Removal of redundant data with the reckoned occurrence*

Begin

1. *Initializing the leaf nodes of a tree with IoT data for each region.*
2. *The hash code h is created for each leaf node using SHA-3 512 bit*
3. *If hash (leaf node) matches with the existing hash (leaf node)*
   then
   a. *Count the occurrence of value by incrementing 1*
   b. *Stores the occurrence value along with the hash value of leaf node in an internal node*
   c. *Remove the redundant data.*
   else
   a. *Allocate the leaf node to store the data*
   b. *Generate the hash for leaf node with occurrence count as 0*
4. *Estimate the assessment of the parent node by hashing the existing node with its adjacent parameter.*
5. *Repeat step 4 until Merkle root is found*
6. *End*

The leaf nodes of the Merkle tree are filled with observed values of IIoT data. The leaf node with hash value of its own is determined and stored in the internal node. If the hash value of new data matches with existing data, then the existence of such value is updated with one step increment. Meanwhile, the redundant data is restricted to enter into the Merkle tree and the hashes of the internal node are determined that reaches the single top Merkle root. The indexing starts from Merkle root to leaf node with an amount of existence of such value for the period of time.

4. Determination of Optimal path in Cloud Environment

Typically in cloud storage, the clustered storage nodes lie in the same zone. Each cluster has one or more nodes that compute resources but does not meet the Industrial requirement with respect to the availability of storage space. To address the above issue, the proposed system uses Linear Programming Problem (LPP) model through which the optimal cost to reach the space availed storage node is calculated. Each node that exists in the cluster is configured with the availability of storage space. Then the space availability at the storage node is compared with the industry demand. Using Vogel's approximation and modified distribution method, the optimal transportation cost is determined, which leads to storing of a huge amount of IIoT with optimal cost in the cloud environment. Initially, resource availability should be confirmed as a balanced or unbalanced problem. If $s_{ij}(\geq 0)$ represents the number of nodes with the availability of storage space from the ith source to the jth destination where the industrial demands storage space, the equivalent Linear Programming Problem (LPP) is given as

$$Z = \sum_{i=1}^{m} \sum_{j=1}^{n} a_{s_{ij}} \, u_{s_{ij}} \tag{4}$$

	d_{u_1}	d_{u_2}	d_{u_3}	d_{u_4}	Source
a_{s_1}	x_1	x_2	x_3	x_4	S_1
a_{s_2}	y_1	y_2	y_3	y_4	S_2
a_{s_3}	t_1	t_2	t_3	t_4	S_3
Demand	d_1	d_2	d_3	d_4	$\sum_{i=1}^{m} a_{s_i} = \sum_{j=1}^{n} u_{s_j}$

If the storage space met the demands of the industry, then it is considered as a balanced transportation problem which is represented as

$$\sum_{i=1}^{m} a_{s_i} = \sum_{j=1}^{n} u_{s_j}$$

(5)

If the availability of storage space does not meet the industry demand, it will result in an unbalanced transportation problem. In such cases, the availability of storage space should be increased to provide resources according to the demands of the industry. The unbalanced problem can be represented as follows

$$\sum_{i=1}^{m} a_{s_i} \neq \sum_{j=1}^{n} u_{s_j} \tag{6}$$

For a feasible solution to the existing problem, the total capacity must be equal to the total requirement. If the total availability of space equals the total demand, then it is a balanced case. As the cloud environment provides abundant storage space, the proposed system assumes that the space available in the storage nodes is always greater or equal to the demand of the customer or industry. With Vogel's approximation method, the industry demand is identified, and the storage node which has excess or equal space is allocated. Similarly, several iterations are carried out to fulfill all the demands of the industry by providing appropriate storage space. Finally, with a modified distribution method, the optimal transportation cost is determined, which results in the storage of a huge volume of industrial data with optimal transportation cost.

Table. 3 Representation of Attributes

Attributes	*Representation*
s_{ij}	*Storage space*
$a_{s_{ij}}$	*Availability of storage space*
$u_{s_{ij}}$	*Industrial demand for storage space*
S_1	*Storage node 1*
a_{s_1}	*Availability at storage node*
d_{u_1}	*Demand of industry1*
C	*Constant*
c_{ij}	*Transportation cost*
d_{ij}	*Degeneracy*

The representation of attributes used for the estimation of transportation cost is mentioned in **Table. 3.** Through this linear programming problem, the minimal cost of distributing the space availed in the storage nodes to the IIoT environment is calculated. If $\sum a_{s_i} = \sum d_{u_j} = C$, the balanced state exists between the available storage nodes and the industrial demand. Then the diff (min (a_{s_i}), next_min (a_{s_i})) is intended with an assignment of penalty. Similarly, the transportation cost for each cell is evaluated. Finally, the degeneracy is estimated from the final solutions. If the number of final solutions contains a value less than $m + n - 1$, then there exists a degeneracy leading to the completion of the optimal cost determination process. Otherwise, the non-degenerate feasible solution is redefined using the modified distribution method with the assignment of $\in (\approx 0)$ in a suitable independent position. Finally, the d_{ij} examined for the cell doesn't contain either $\sum a_{s_i}$ or $\sum d_{u_j}$

. $$d_{ij} = c_{ij} - (u_i + v_j) \qquad c_{ij} \ni \sum a_{s_i} \; or \; \sum d_{u_j} \tag{7}$$

From the above estimation, it is concluded that if $d_{ij} > 0$, then the determined cost is optimal and unique. If $d_{ij} = 0$, the cost is optimal, which facilitates an alternative optimal solution. If $d_{ij} < 0$, the solution is not optimal, and hence the iteration of the calculation process repeats to attain the optimal state. With the above calculation, the proposed system determines the optimal transportation cost to store the non-redundant data in the cloud environment.

5. Security Analysis at proposed system

Each node in factory is assigned with unique identity, shared secret key using AES algorithm in predeployment phase. The nodes in the Industrial environment generates shared session key to guarantee the data transmission with edge server and other nodes. When the edge server receives data packages it decrypts the package before get processed. During the outsourcing of other information to the cloud it is again encrypted with AES algorithm to prevent from data leakage. The cloud accumulates a huge volume of data produced in the Industrial environment. On the other hand, the data accumulated in the structure of ciphertext, which is not deciphered devoid of the data users' undisclosed inputs [21]. Additionally, the Merkle tree contains hashes of hashes for the data stored in the leaf node through which the data will not be corrupted. Only if the chain is followed continuously, the real data may be collected at the end. Failure results in the processing of tampered data.

6. Performance Evaluation

To assess the performance of proposed scheme in a cloud environment, experimental setup has been arranged as mentioned **Table.4.**

Table. 4. Experimental Setting

Parameter	Value
Number of nodes	10
Edge server	Machine-I Laptop (AMD Ryzen 7 4700U Processor, 2.1GHz processing speed and 8 GB RAM, Windows 10)
Cloud server	Machine-II Desktop (Intel Xeon E5 2620 v2 workstation , 2.1GHz processing speed, 64GB RAM , Centos 7)

Cloud platform	Eucalyptus 4.3.1v
Cloud type	Private Cloud
Connection of the nodes	Zigbee
Connection between edge and cloud server	The Internet
Storage Controller	Walrus (W)
Sensing Interval	10s

6.1 Experimental Setup

In this segment, the proposed system performance are evaluated based on the real experiments. The proposed system builds the prototype to measure the distance of the liquid level in the containers as well the temperature of the factory environment. In this experiment, sensor producing same kind of output at various stages is considered. The proposed system is examined with various factors like space reduction percentage, efficient data retrieval, network lifetime, time series analysis, average latency, data transmission and storage space

In the experiment, five ultrasonic sensor and five temperature sensor nodes are deployed in the factory to build IIoT environment. The sensors nodes are connected via wireless mode of communication to observe the environment. The sensor nodes employed for the experiment were made communicated with each other through Zigbee protocol. Zigbee (IEEE 802.15.4) is a low-power, low-data rate wireless network and enables smart objects to work securely on any network. Laptop of following configuration (AMD Ryzen 7 4700U Processor, 2.6 GHz Intel Core processor, Windows OS 10 and RAM of 8 GB) is employed to communicate with sensor nodes. The data transfer between edge server and cloud server carries with the help of Internet. The edge server (Machine –I) receives the data then transmits to a cloud. Machine-II is employed as cloud server in the proposed prototype. The distance value and temperature value measured by the sensor nodes are collected through Arduino module which functions at the edge server. Finally the recorded value is outsourced to the cloud server for intelligence actions. The reading are recorded every ten seconds and fusion of recorded values are passed to cloud every hour. Since, ten sensor nodes are deployed over factory, ten reading were recorded every ten seconds. To compute time series data from collected IIoT data, Index based deduplication using Merkle tree is proposed in this research work. To carry out lingering proposed work, cloud environment is built with Eucalyptus. Eucalyptus is a private cloud contains Cloud Controller to execute administrative process and Walrus, the storage controller to accumulate an enormous amount of IIoT data produced by the IoT devices embedded in the industrial environment. The partitioning of regions in the collected sensor values and de-duplication using the Merkle tree is implemented using python. Finally, the optimal transportation cost to store data in cloud is determined with the available node information at the storage controller.

6.2 Space Reduction due to de-duplication

Improving the efficiency of cloud storage by de-duplication is one of the main intentions of the proposed system. To measure the performance of the proposed system, a total of 10 sensor nodes values were considered and each node observes approximately 1, 30,000 of sensor values. The sensor values are partitioned into region and sub region based on time period and boundary value. Using the Merkle tree the sub region sensor values are indexed with generation of SHA 3 512 bit hash value respectively. Under each node sensor values, a range of 90 to 120 regions were formed based on timestamp and each region holds approximately 14 sub regions. On performing various experiments it is observed that 90% of sensor values on all regions were similar under each node information. Furthermore, the de-duplication carried at one region with the proposed scheme creates much impact on other regions by maintaining the single instance with computation of existence at each region. The De-duplication Ratio is calculated by dividing the total number of sensor values obtained after de-duplication process by the total number of sensor values considered for de-duplication.

$$DR = \frac{No.of\ sensor\ values - No.of\ sensor\ values\ obtained\ after\ deduplication}{No.of\ sensor\ values}$$

Table.5 Performance of Proposed work

Node	Memory Size (KB)		Total No. of Regions	Total No. of Sub-regions	% of values reduced per region	% of values reduced per node
	Before De-duplication	After De-duplication				
1	4093	359	97	10	98.6	99.2
2	3248	342	116	14	97.9	98.7
3	3358	336	117	15	98.4	99.1
4	3295	316	118	14	98.2	98.9
5	3346	326	121	16	97.9	98.5

The de-duplication process carried out with the partition of regions and sub-regions is depicted in **Table.5.** As the results attained towards de-duplication system and space reduction percentage presented in **Fig. 3** and **Fig. 4,** proves that 10% of data under each node sensor values is unique and remaining values are identified as redundant data and the occurrence count is measured to provide time series data.

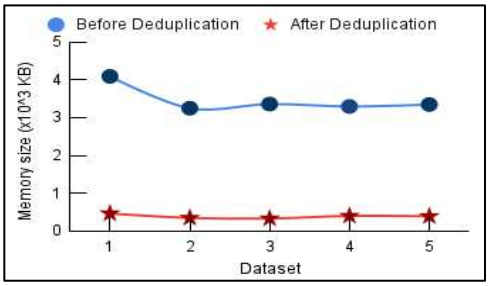

Fig. 3. Deduplication of Proposed Scheme

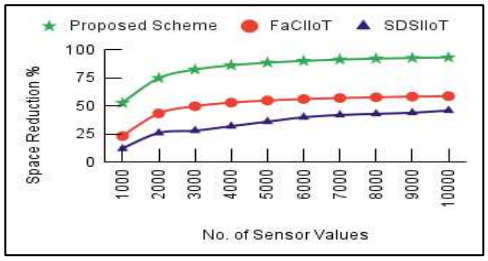

Fig. 4. Space Reduction Percentage

6.3 Efficient Data Retrieval

The proposed work provides a simple approach to de-duplicate the redundant IIoT data and it can be retrieved from the cloud server with minimal time. The proposed system performance is measured with FaCIIoT [20] and the SDSSIIoT [21] to establish the average data retrieval time on the cloud server. In FaCIIoT, it requires more time for scanning to obtain the results from the query. The results increased linearly with an increase in the feature vectors. However SDSSIIoT gave good results in the data retrieval mechanism, many relevant data were produced along with explored data. Whereas in the proposed system, the Merkle Hash Tree gave the best performance in the data retrieval mechanism with exact data as it maintains a single instance for the each node sensor values. The proposed scheme maintains a unique structure for the indexing methods with an additional feature of reckoning the occurrence to count the existence of each value. Due to several hits made by the redundant data to acquire space in the cloud storage, paths established between root and data become recurrent. Thus, the search proportion gradually decreased, which enabled good results, as revealed in **Fig. 5.**

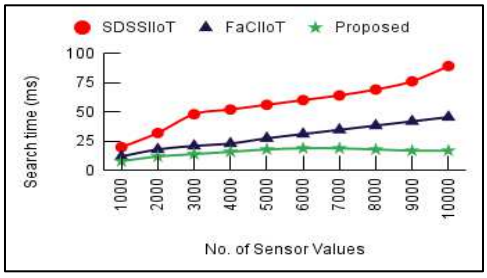

Fig. 5. Search Time

6.4 Network Lifetime

The improvement of the network lifetime using the proposed system is a demanding responsibility. In this framework, continuance is evaluated with FaCIIoT [20] and the SDSSIIoT [21]. It is evident from the plots that the proposed system attains an elevated network lifetime when compared to the previous works. The average network continuance obtained in the preceding researches was lesser than the proposed system value of 60s. In SDSSIIoT the network lifetime was not much good because the authors have focused only on the secure storage scheme for IIoT data. Even many de-duplication systems exist to address the IoT data, de-duplication carried out with indexing-based technique is the newest approach. In the FaCIIoT, the network lifetime was achieved with good value. But it failed to persist for longer durations. Attaining maximum network lifetime requires balancing of traffic overhead between sensor nodes and cloud storage. The proposed system maintains a good network lifetime for a longer duration which is shown in **Fig. 6.** It is due to the high availability of storage space at the cloud server and reduction of network traffic overhead by evading redundant data to reach storage. The cloud persisted in receiving data from proxy servers due to the availability of resources.

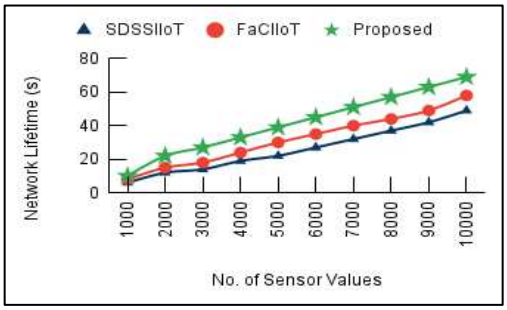

Fig. 6. Network Lifetime

6.5 Computation time for decision making

Figuring time–series analysis concurrently during the de-duplication process in the cloud storage is one of the challenging tasks which is addressed by the proposed work. The performance of the proposed work is compared with existence works like FaCIIoT [20] and the SDSSIIoT [21]. Merely ten node readings were taken into consideration for evaluating this parameter. The average time taken by the proposed system to execute the decision-making task is 47 ms and the resilience for consuming less time in decision making is plotted in **Fig. 7.** Whereas SDSSIIoT takes 88ms and FaCIIoT takes 107ms. The proposed system consumes less than 50% of time for decision making process when compared to previous works.

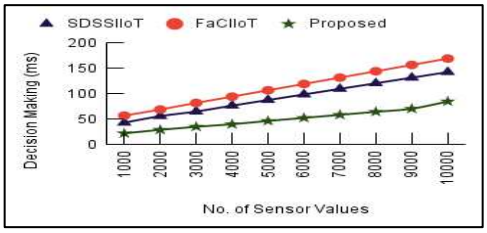

Fig. 7. Comp. Time of decision Making process

6.6 Average Latency

The proposed work used categorized regional method and Index based de-duplication methods to reduce the redundant data in spite of its quick user, which achieved excellent latency time. The latency time is the duration needed to react to the customer's needs. The mean latency is characterized by the addition of the duration to fulfil every demand put forward by the user. It is calculated based on the lesser and greater value of the duration. The lesser duration is 0, whereas the greater duration is the interval needed for assimilating the user's solitary demand. The average latency is calculated using the previous works, which are the FaCIIoT [20] and SDSSIIoT [21]. Which are considered for determining the effectiveness of the proposed system. As presented in **Fig. 8**, the mean latency for 5000 sensor values using the proposed system produces 1.46ms, which is lesser than the FaCIIoT, SDSSIIoT.

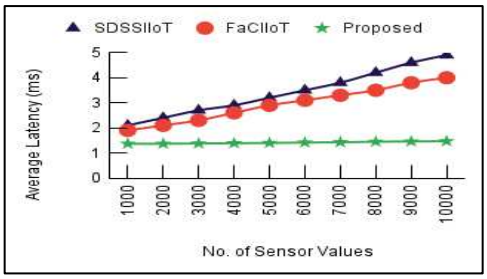

Fig. 8 Average Latency

6.7 Data Transmission Rate in the Cloud Server

The sensor nodes connected in the factory measure the distance value and temperature value every 10s then sends the observed value to the edge server. The collected value of every one hour is passed to cloud server. The proposed focuses on determining optimal path to transmit the data in cloud server. As obtained in **Fig.9**, the proposed system greatly reduces the transmission rate. This can be described by the fact that the raw data observed from the sensor nodes is preprocessed before outsourcing to the cloud storage as well the optimal path in the cloud storage is identified with the storage node information available at Walrus. Therefore, with this proposed framework, the cloud needs to store only one reading instead of five readings. As the proposed system computes the time series during the deduplication of IIoT data, only less amount of data is transmitted when compared to previous works like FaCIIoT [20] and SDSSIIoT [21].

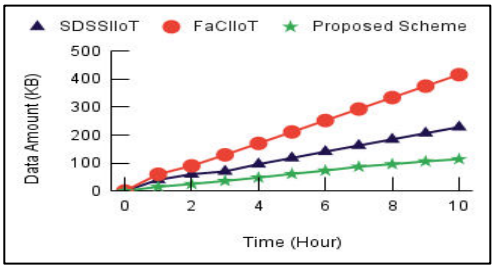

Fig. 9 Data Transmission

7. Conclusion and Future works

The data procured by the industries has augmented enormously due to the development of automation in Industries. Extensive comparable data in the repository server exists, which needs to be avoided. Despite the problems exist with various existence de-duplication schemes, the proposed system is designed with additional feature called reckoning of occurrence in the multilevel indexing Merkle tree. To compute time series data, the proposed system implements Categorized regional method to partition the data based on time interval which creates much impact during decision making process. Finally the optimal transportation cost to reach the storage node in the cloud also addressed. From the experimental results, the proposed system was revealed to perform in an enhanced manner when compared to the previous works in terms of space reduction percentage, search time, network lifetime, decision making process, average latency and data transmission rate. There are still many challenges in the implementation of this proposed system. Open research based on security needs to be implemented for improving this framework efficiency. In future, we have planned to concentrate on other applications like IoT healthcare services at edge environments.

Declarations

Funding: Not Applicable

Conflicts of interest/Competing interests: Not Applicable

References

[1] Hongming Cai, Boyi Xu, Lihong Jiang, Athanasios V. Vasilakos, IoT-Based Big Data Storage Systems in Cloud Computing: Perspectives and Challenges, *IEEE Internet of Things Journal,* Vol. 4, No. 1, pp. 75-87, February, 2017.

[2] Industrial IoT Market – Global Opportunity Analysis and Industry Forecast (2020-2027) Report, 2019. *https://www.meticulousresearch.com/product/industrial-iot-market* Accessed Jan, 2021

[3] Rajalakshmi Krishnamurthi, Adarsh Kumar, Dhanalekshmi Gopinathan, Anand Nayyar, Basit Qureshi, An Overview of IoT Sensor Data Processing, Fusion and Analysis Techniques, *Sensors*, *MDPI*, Vol. 20, No.21, pp.6076, October, 2020.

[4] Lixing Chen, Pan Zhou, Liang Gao ,Jie Xu , Adaptive Fog Configuration for the Industrial Internet of Things, *IEEE Transactions On Industrial Informatics*, Vol. 14, No. 10, pp. 4656-4664, October, 2018.

[5] Geetanjali Rathee , Sahil Garg , Georges Kaddoum , and Bong Jun Choi , Decision-Making Model for Securing IoT Devices in Smart Industries, *IEEE Transactions on Industrial Informatics*, Vol. 17, No. 6, pp. 4270-4278, June, 2021.

[6] E.M. Borujeni, D. Rahbari, M. Nickray, Fog-based energy-efficient routing protocol for wireless sensor networks, *Journal of Supercomputing*, Vol. 74, No. 12, pp. 6831-6858, December, 2018.

[7] G. Peralta, P. Garrido, J. Bilbao, R. Aguero, P.M. Crespo, On the combination of multi-cloud and network coding for cost-efficient storage in industrial applications, *Sensors*, *MDPI*, Vol. 19, No. 7, pp. 1-19, April, 2019.

[8] Priteshkumar Prajapati, Parth Shah, A Review on Secure Data Deduplication: Cloud Storage Security Issue, *Journal of King Saud University – Computer and Information Sciences*, November, 2020.

[9] K. Akhila, A. Ganesh, C. Sunitha, A Study on De-duplication Techniques over Encrypted Data, *Fourth International Conference on Recent Trends in Computer Science & Engineering, Procedia Computer Science*, Thrissur, Kerala, 2016, pp.38-43.

[10] Xiaoyu Zheng, Yuyang Zhou, Yalan Ye, Fagen Li, A cloud data de-duplication scheme based on certificateless proxy re-encryption, *Journal of Systems Architecture*, Vol. 102, pp. 101666, January, 2020.

[11] Wen Xia, Dan Feng, Hong Jiang, Yucheng Zhang, Victor Chang, Xiangyu Zou, Accelerating content defined-chunking based data de-duplication by exploiting parallelism, *Future Generation Computer Systems*, Vol. 98, pp. 406-418, September, 2019.

[12] Yinjin Fu, Nong Xiao, Hong Jiang, Guyu Hu, and Weiwei Chen, Application-Aware Big Data Deduplication in Cloud Environment, *IEEE Transactions on Cloud Computing*, Vol. 7, No. 4, pp. 921-934, October, 2019.

[13] Wen Xia, Hong Jiang, Dan Feng, and Lei Tian, DARE: A Deduplication-Aware Resemblance Detection and Elimination Scheme for Data Reduction with Low Overheads, *IEEE Transactions on Computers*, Vol. 65, No. 6, pp. 1692-1705, June, 2016.

[14] Zheng Yan, Wenxiu Ding, Xixun Yu, Haiqi Zhu, and Robert H. Deng, Deduplication on Encrypted Big Data in Cloud, *IEEE Transaction on Big Data*, Vol. 2, No. 2, pp. 138-150, June, 2016.

[15] Chia-Mu Yu , Sarada Prasad Gochhayat , Mauro Conti , and Chun-Shien Lu, Privacy Aware Data Deduplication for Side Channel in Cloud Storage, *IEEE Transactions On Cloud Computing*, Vol. 8, No. 2, pp. 597-609, April, 2020.
[16] J. Ni, K. Zhang, Y. Yu, X. Lin, X.S. Shen, Providing task allocation and secure de-duplication for mobile crowdsensing via fog computing, *IEEE Transaction on Dependable Secure Computing*, Vol. 17, No. 3, pp. 581-594, May, 2018.
[17] Guohua Tian, Hua M, Ying Xie, Zhenhua Liu , Randomized de duplication with ownership management and data sharing in cloud storage, *Journal of Information Security and Applications*, Vol. 51, pp. 102432, April, 2020.
[18] Shunrong Jiang, Tao Jiang, and Liangmin Wang, Secure and Efficient Cloud Data Deduplication with Ownership Management, *IEEE Transactions on Services Computing*, Vol. 13, No. 6, pp. 1152-1165, November, 2020.
[19] Yuan Gao, Hequn Xian and Aimin Yu, Secure data deduplication for Internet-of-things sensor networks based on threshold dynamic adjustment, *International Journal of Distributed Sensor Networks*, Vol. 16, No. 3, pp.155014772091100, March, 2020.
[20] Shivi Sharma, Hemraj Saini, Fog assisted task allocation and secure de-duplication using 2FBO2 and MoWo in cluster-based industrial IoT (Industrial IoT), *Computer Communications*, Vol. 152, pp. 187-199, February, 2020.
[21] Jun-Song Fu , Yun Liu , Han-Chieh Chao ,Bharat K. Bhargava, Zhen-Jiang Zhang, Secure Data Storage and Searching for Industrial IoT by Integrating Fog Computing and Cloud Computing, *IEEE Transactions on Industrial Informatics*, Vol. 14, No. 10, pp. 4519- 4528, October, 2018.
[22] Manogar Ellapan, Abirami S, Dynamic Prime Chunking Algorithm for Data Deduplication in Cloud Storage, Vol. 15, No. 4, pp. 1342-1359, *KSII Transactions on Internet and Information Systems,* April, 2021.
[23] R. Veerachamy, V. Ravi Kumar, " Operational Research", I K International Publishing, 2011.