Data Classification and Compression for Efficient Sensor-Cloud Communication

[1]Vijay Maheshwari,
[1]Shobhit Institute of Engineering and Technology (Deemed to be University), Meerut
Email Id- [1]vijay@shobhituniversity.ac.in,

Abstract: The Wireless Sensor Network (WSN), which consists of a collection of specialized sensors connected by a communication infrastructure for monitoring and managing conditions in many places, is a relatively new technology that is gaining in popularity. Furthermore, cloud computing is a kind of high-performance computing that makes use of a network of distant servers to store, manage, and analyze data instead of a local server or personal computer. By integrating the capabilities of both ends, sensor-cloud architecture is also delivering excellent services. To offer such services, a significant volume of sensor network data must be transmitted to a cloud gateway, which necessitates a considerable quantity of bandwidth and time. In this article, we present an effective sensor-cloud communication method that uses statistical classification based on machine learning, as well as compression utilizing the deflate technique with minimum information loss, to reduce the huge bandwidth and time requirements. The experimental findings describe the suggested method's overall efficiency in comparison to conventional and related research.

Keywords: Wireless Sensor Network, Cloud Computing, Classification, Compression, Sensor-Cloud Communication.

1. INTRODUCTION

Cloud computing and wireless sensor networks (WSN) are two new technologies that have gained a lot of traction in the information and communication technology sector. Because of problems such as restricted bandwidth, unstable channels, heterogeneity, and so on, wireless storage systems as part of wireless networks have not been utilized in large-scale applications. These flaws are overcome when integrating WSN with the cloud environment. The storage restriction of sensors in sensor devices is also addressed by adding compression techniques. The amount of data and transmission energy are decreased using these methods. Data compression, on the other hand, will be substantial if the algorithm's execution does not use more energy than each transfer. The solution to this issue is to move the compression operation entirely to the gateway. The authors of suggested a paradigm in which WSN and cloud communication is based on gateways. Data is gathered at each sensor (deployed in different locations) and transmitted straight to the sensor gateway without any sensor-end processing. The gathered data is fed into a neural network, which is subsequently compressed to reduce transmission time. The power consumption at the sensor node is minimized since the processing is done at the sensor gateway [1]. The time between each successive data collection should be maintained as short as possible to improve accuracy.

As a consequence, the data size is rapidly increasing. Because the time intervals are equivalent, there will be a lot of identical data after completing a cycle (e.g., a month or a year) due to seasonality. When the environment does not change quickly, data duplication happens once more. In this study, we looked at the fundamental features of sensor-cloud infrastructure, as well as the suitable communication medium and network design. We've also highlighted the main difficulties of sensor-cloud communication, including a difficulty with real-time sensor-cloud apps using a lot of bandwidth. In light of this, we've developed an effective communication framework to address the bandwidth problem, thus increasing the 10 Efficient Sensor-Cloud Communication Using Data Classification and Compressiona sensor-cloud architecture's efficiency We have created a sample application and compared the conventional and suggested frameworks, as well as relevant studies. To our knowledge, this is one of the first papers to carefully investigate the bandwidth requirements for sensor-cloud communication. Second, by combining the idea of machine learning with data categorization and compression methods, we have suggested how to decrease bandwidth usage with minimum information loss during transactions between sensor networks and cloud environments. Finally, we evaluated the time needed for our framework and compared it to the conventional method. Finally, with related studies, we manually evaluated the total bandwidth usage and time need. The following is how the rest of the paper is organized: An overview of sensor-cloud architecture is given in section II. A overview of similarity-based categorization is given in section III. In section IV, the suggested work's issue statement is described. The suggested framework is presented in

Section V, along with an activity diagram. Section VI depicts a thorough study and execution of the framework. Finally, a comprehensive description of the simulation findings is given in section VII. The planned work is summarized in section VIII, which includes closing remarks. II. Sensor-Cloud architecture integrates WSN with cloud computing in a manner that allows for real-time data processing and storage of sensor network data, as well as analysis of the processed data to disclose hidden sights. This integrated infrastructure may be thought of as a cloud computing extension that can handle WSN physical sensors to satisfy the growing need for large-scale wireless network applications.

A wireless sensor network (WSN), also known as a wireless sensor actuator network (WSAN), is a collection of autonomous sensors that are geographically dispersed. The computational power and storage capacity of these sensors are usually modest. These small sensors can detect, measure, and collect information from the environment for any monitoring and controlling application, and then communicate the data to the user. In most cases, a WSN does not have any specific infrastructure. As a consequence, it may be divided into two categories: organized and unstructured. Physical sensors are installed ad-hoc in an unstructured WSN, while in a structured network, there must be a pre-plan for installing all or part of the sensor nodes. Weather forecasting, military command and control natural disaster relief management, e-health, and so on are examples of wireless sensor network applications. Cloud computing, often known as on-demand computing, is an internet-based computing model that allows users to obtain on-demand services through pooled processing resources. The word "cloud" is a metaphor for "the internet" in cloud computing. As a consequence, the customer receives all cloud services (such as software, platform, and infrastructure) through the internet. Cloud computing may be conceived [2] of as a network computing paradigm in which the servers are virtual machines or real computers in the cloud. It depends on resources being shared across various cloud services to achieve coherence. Cloud computing also offers a fantastic and easy user experience since end users don't have to worry about where the servers are physically located. They may access the service by utilizing a login panel to connect to the server. The WSN is connected with the cloud environment in sensor-cloud architecture to enable easy processing and storage. This method enables the sensor network to collect and send all sensor data to the cloud at regular intervals. Figure 1 shows the Cloud Computing [3].

Figure 1: Cloud Computing.

2. DISCUSSION

2.1. Application:

The sensor-cloud infrastructure (i.e., the combined WSN and cloud computing infrastructure) is a one-of-a-kind sensor data storage, analysis, and monitoring platform that employs a scalable cloud computing method to provide superior data analysis and visualization. The constraints of WSN, such as restricted storage, processing, and power consumption, are addressed using this method. Because cloud computing offers a lot of storage and processing power, it allows the sensor network to gather a lot of data by connecting it to the cloud through gateways. The sensor gateway gathers sensor network data and compresses it before sending it over the internet to the cloud gateway. The goal of a similarity and distance measure is to compare two sets of data (records and vectors) and compute a single number that indicates how similar they are[4]. To comprehend the proximity characteristic of two data sets, these measurements are required. However, selecting an adequate similarity measure is crucial for categorization. In classification and clustering, there are many methods to calculate the distance between distinct patterns of data sets. The following are a few of them: Euclidean Distance: A particular instance of Murkowski distance is Euclidean distance. It's the distance between two locations in two-dimensional space measured in straight lines. If (in Cartesian coordinates as well as if there are two points, the distance between them is, if there are two points, the distance between them is, if there are two points, the distance between them is, if there are two points, the distance between them is, if there are two Cosine Similarity: Cosine similarity is a kind of similarity measure that estimates the cosine of the angle between two items. It evaluates the judgment of orientation rather than magnitude (i.e. the cosine similarity of two vectors with the same orientation is 1, the similarity is 0 if the angle between them is 90°, and the similarity is -1 if the vectors are diametrically opposite of each other, regardless of their magnitude. In non-negative space, cosine similarity is very useful, as the result falls inside the range [0,1]. Given two attribute vectors, and, the conise similarity, is represented by a dot product with magnitude as. Classification, which is a supervised learning method in statistics, is one of the processes for categorizing a new collection of records into which of a specified class it belongs based on a training data set with known category membership.

Various techniques, such as machine learning, may be used to identify the preset classes. Machine learning is a technique for creating sophisticated models that can make predictions in the absence of a training set of properly specified data These analytical models may assist us in drawing accurate conclusions and uncovering hidden insights from data patterns. Lossless compression is a kind of data compression that enables compressed data to be decompressed without losing any information. Deflate is a lossless data compression method that combines the LZ77 and Huffman coding algorithms to compress data efficiently. The Deflate method is used by a number of free and open source data compression programs. Bluetooth or Wi-Fi [5] may be used to communicate between the WSN and the sensor gateway. The connection between the cloud gateway and the client, on the other hand, is crucial. It is possible to have a wired or wireless cloud environment. The communication between the parties is the most challenging aspect. a large-scale sensor gateway and cloud gateway Data will be provided on a regular basis. Since the internet has become so popular, It is regarded as a communication medium, and it requires a excessive traffic due to a large quantity of bandwidth transmission, high internet costs, and data security are all issues. Sensor networks, as we all know, detect data. We may anticipate a lot of things in a regular time span.

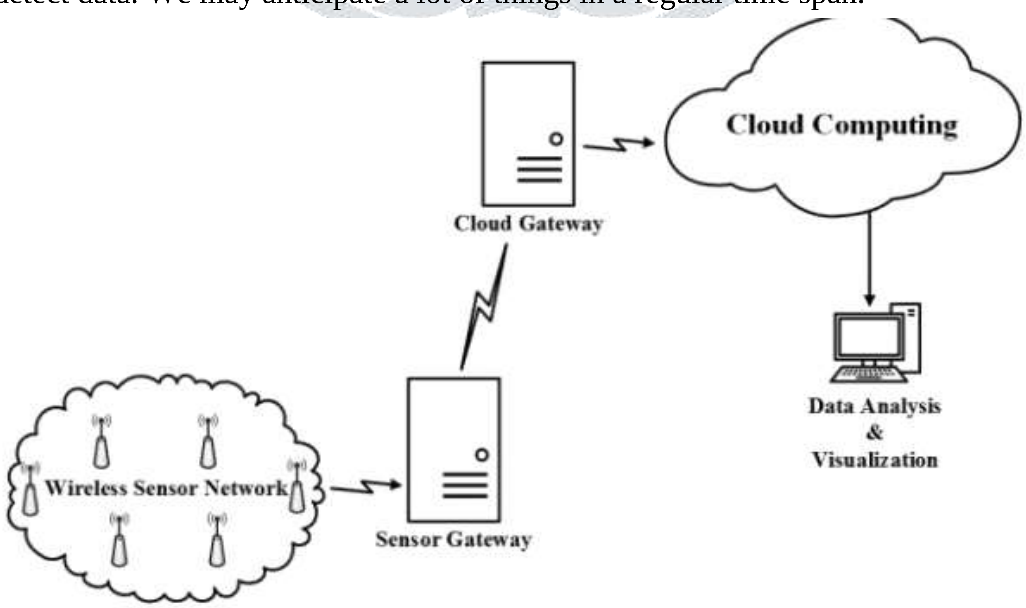

Figure 2: Sensor-Cloud infrastructure.

Commonalities in the same cycle at the same time period (since the data is usually collected over a long period of time) Observe the seasons. Instead of delivering similar messages, we suggest sending documents (a set of sensor values in a specific time interval) We prefer to transmit a single code over and over again. will reflect the whole album, and before delivering it, we'll make sure it's perfect. a compression method (deflate algorithm) is also used, lowering the total bandwidth required. The suggested framework's process is shown in the diagram below. To begin, we'll need a Look Up Table (a collection of names). preset records with unique codes) that will be used tousled to determine the degree of similarity The Look Up Table is a table that may be used to look up information. be constructed by mining the data (using a machine learning technique) previous data throughout the whole cycle The Look Up Table must be accessible at this stage. both entrances (sending and receiving). For the first time, it should be compared to each of the company's records. Look Up Table. If a certain level of resemblance is required, instead of sending the sum, the matching code is sent. record. The records that aren't the same will be included to the mix. Look Up Table, and at the end of each cycle, the frequency of the Look Up Table should be updated. in terms of hit ratio Figure 2 discloses the Sensor-Cloud infrastructure [6].

2.2. Working:

We may anticipate a lot of similarities in the same time period of the different cycle (a full period of time) since the data usually follow seasonality, as we know sensor networks receive data in a periodic time interval. In our proposed solution, instead of sending similar records (a set of sensor values in a specific time interval) again and again we prefer to send a single code which \swill represent the entire record and before sending it we sare also using a compression algorithm which will reduce overall bandwidth requirement. The workflow of the proposed framework is shown in. First and foremost, we'll need a Look Up Table (a set of preconfigured entries with unique codes) to calculate the similarity. The Look Up Table can be created by [7] mining (using machine learning method) the prior data for a full cycle. At this stage, the Look Up Table needs to be accessible at both gateways (sending and receiving). A new record should be compared to every other record in the Look Up Table. If a standard amount of similarity is discovered, instead of sending the whole record, the matching code is delivered. Dissimilar records will be added to the Look Up Table, and the Look Up Table will be updated after each cycle based on the frequency of hit ratio. We used environmental data obtained at one-minute intervals from six sensors (Air Temperature, Dew Point, Humidity, Pressure, Wind Speed, and Sea Level Pressure) in our study. For a one-year period, we received more than 5 lace records from the data source.

A. Seasonality detection using time series analysis We consider the data to be a time series since it is gathered at a defined time period and must have seasonal and [8] cyclical impacts. We're treating a single day as a cycle, with each hour representing a season. As a consequence, a single cycle will have twenty-four seasons. We computed seasonality for each month to isolate the total seasonal impact of a year (full cycle). From February to October, the graph of seasonal indices follows a path from low to zenith and then rests at low again, as shown in. Another trend with numerous ups and downs occurs from November through January. We may now infer that seasonality occurs within consecutive months. We have also discovered the similarities between the subsequent months, since our primary goal is to identify similarity among each record in the data. As a consequence, we may anticipate that, despite similarity within each month, there will be similarity among the others. As a result, the Look Up Table was created using a machine learning methodic. Measure of Similarity We've classified a record as a Look Up Table record if the Euclidean distance between it and every other record is smaller than a certain threshold, k. Here, k is the maximum allowable Euclidean distance, which is calculated by adding all of the standard deviations. We identified 1585 entries in the Look Up Table using equation (3), which categorized approximately 98 percent of the entire data with an excellent hit ratio. We computed the Euclidean distance between each record and every other record. If the minimum distance is less than (threshold value), the record is added to the Look Up Table as a new item. We have the Look Up Table for a full cycle at this point, which will be utilized to categorize fresh entries. When a new record arrives in the following cycle, it will be compared to all Look Up Table records. If similarity (as determined by the previous similarity measure) is discovered, just the appropriate code is transmitted to the cloud gateway, and the frequency for that specific record is updated. Otherwise, the raw record was transmitted to the cloud gateway unaltered and is now deemed a new Efficient Sensor-Cloud Communication with Data Classification and Compression. Look Up Table is a record in which you can look up information about a table. We now have an updated Look Up Table with various hit rates after completing the cycle. We have removed the least utilized entries from the table for the next full cycle.

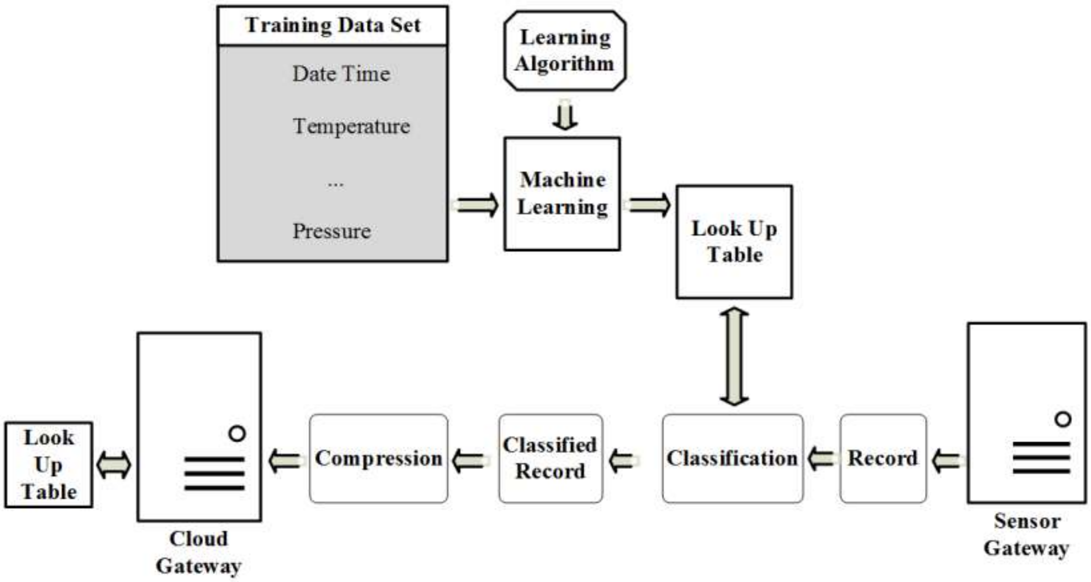

Figure 3: Flow diagram of proposed framework

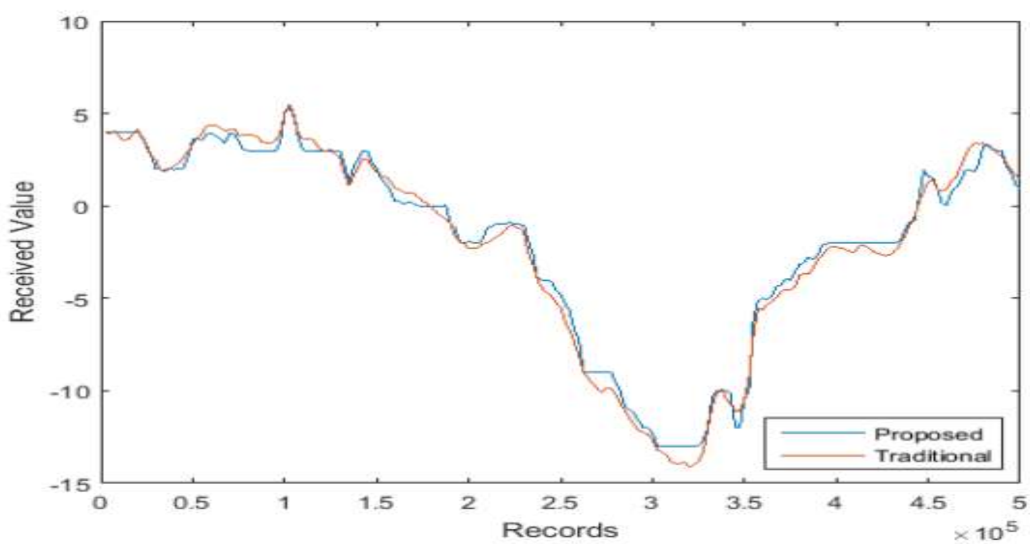

Figure 4: Residual Measure of Received Data

To assess the suggested framework's performance, we ran the necessary simulations for various situations, and the results were compared to the conventional method and. Our research's main goal was to see whether there [4]was a way to reduce bandwidth use in sensor-cloud connection. As a result, we estimated the necessary bandwidth for each transmission and discovered a substantial difference between the conventional and suggested approaches. We also examined the total bandwidth for both instances (with and without compression), as well as the conventional method. Even though we only examined a small number of sensors (six in our instance), the effect will become more apparent as the network grows. In virtually every form of communication, the time it takes for a message to finish is critical. As a result, a framework maybe called efficient if it consumes less bandwidth and processes data in a reasonable period of time. We used a 128 Kbps connection between the sensor and the cloud gateway in our study. We may claim that any bytes will need sec or millisecond based on equation 4. As a result, it's obvious that as bandwidth grows, so does the time it takes to transmit data. We measured the processing and transmitting times for each record in our proposed architecture. The cumulative time comparison between the conventional and suggested approaches is shown in the graph below. Figure 3 discloses the Flow diagram of proposed framework [7].

3. CONCLUSION

Sensor-cloud is a popular combo in recent years because it offers a new framework for speeding service innovation and cross-disciplinary applications that cut across organizational boundaries. Communication between WSN and cloud computing in sensor-cloud architecture is a difficult job due to the high bandwidth requirements. Our suggested architecture may decrease the robust [9] bandwidth need of sensor-cloud connection to the point where data loss is minimal and the transaction time is reduced to some degree. In comparison to similar studies, we discovered that our suggested architecture provides 4.7 times higher performance. We will continue to expand our study by examining the suggested framework in a variety of real-world applications in order to improve overall performance. Figure 4 discloses the Residual Measure of Received Data [10].

REFERENCES

[1] C. Zhu, J. J. P. C. Rodrigues, V. C. M. Leung, L. Shu, and L. T. Yang, "Trust-based communication for the industrial internet of things," *IEEE Commun. Mag.*, 2018.

[2] M. T. Rahman, M. S. A. Salan, T. F. Shuva, and R. T. Khan, "Efficient Sensor-Cloud Communication using Data Classification and Compression," *Int. J. Inf. Technol. Comput. Sci.*, 2017.

[3] J. Lindström, A. Hermanson, F. Blomstedt, and P. Kyösti, "A multi-usable cloud service platform: A case study on improved development pace and efficiency," *Appl. Sci.*, 2018.

[4] R. F. Babiceanu and R. Seker, "Big Data and virtualization for manufacturing cyber-physical systems: A survey of the current status and future outlook," *Comput. Ind.*, 2016.

[5] M. S. Hossain and G. Muhammad, "Cloud-assisted Industrial Internet of Things (IIoT) - Enabled framework for health monitoring," *Comput. Networks*, 2016.

[6] T. Le, N. T. Le, and Y. M. Jang, "OCC-ID: New Broadcasting Service-Based Cloud Model and Image Sensor Communications," *Int. J. Distrib. Sens. Networks*, 2016.

[7] J. Zhou, Z. Cao, X. Dong, and X. Lin, "Security and privacy in cloud-assisted wireless wearable communications: Challenges, solutions, and future directions," *IEEE Wirel. Commun.*, 2015.

[8] A. Alamri, W. S. Ansari, M. M. Hassan, M. S. Hossain, A. Alelaiwi, and M. A. Hossain, "A survey on sensor-cloud: Architecture, applications, and approaches," *International Journal of Distributed Sensor Networks*. 2013.

[9] T. Alam and M. Benaida, "CICS: Cloud-internet communication security framework for the internet of smart devices," *Int. J. Interact. Mob. Technol.*, 2018.

[10] R. Singh, E. Singh, and H. S. Nalwa, "Inkjet printed nanomaterial based flexible radio frequency identification (RFID) tag sensors for the internet of nano things," *RSC Advances*. 2017.