ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 5, Issue 4, April 2018

Providing an Efficient Support on Fast Detection and Locating of Errors in Big Sensor Datasets

[1] Harshit Mandada, [2] Malaserene I
[1] B.Tech Student,[2] Associate Professor
[1][2] Department Of Computer science, Vellore Institute of Technology, VELLORE District, Tamil Nadu, INDIA

***Abstract* - Now-a-day's maintaining or processing big data is a critical challenging problem. One of important source for scientific big data is the datasets gathered through the wireless sensor networks (WSN). For a WSN application to realize an appropriate result, it is required that the data received is clean, accurate, and lossless. However, effective error detection as well as cleaning of sensor big data errors is a challenging problem demanding novel solutions. In this paper, we propose a novel error detection approach to detecting errors in big data sets from sensor networks. This approach is a best solution to detecting errors even faster in big sensor data. This proposed approach works based on scale-free network topology and also this approach will be based on the classification of error types. By implementing this approach we can achieve significant time performance enhancement in error detection without compromising error detection accuracy.**

Key words: Big Data, Error Detection, Sensor Network .

I. INTRODUCTION

Big data is a technology used to store as well as process the aggressively enhancing dataset which includes structured, semi structured and unstructured data that has to be searched for precious information. Big data deals with 3 V's: those are, Volume, Variety, Velocity of processing big data. It is associated with cloud computing for the analysis of large data sets in real time. Volume refers to the vast amount of data it collects, Velocity refers to the speed at which it process the data as well as Variety defines that data does not mean just numbers,dates or strings but also geospatial data, 3D data, audio, video, social files, etc. The key purpose of the big data technology is to achieve high level quality of data and convenience for business aptitude. It replaces the traditional warehousing, relational databases and complicates software tools used to progression this huge data. The example of big data is petabytes & exabytes. With the fast development in the networking technology which handles massive data at a time, the big data technology is gaining importance present days. It is now rapidly expanding in all science and engineering domains, together with physical, biological and biomedical sciences. It is also used in government agencies, financial corporations, large enterprises, etc. However, big data technology has various advantages it also imposes challenges such as the data that is stored in big data is from various sources at various rates, hence the date from one source will be out of management from the other source which is to be coordinated. Secondly, the major challenge lies in extracting useful data from the massive stored data at low cost which is known as data mining.

Big data is the term for any group of data sets so large as well as complex that it becomes complicated to process using existing data processing applications. The challenges contain analysis, capture, search, distribution, storage, transfer, visualization, and privacy violations. The trend to larger data sets is due to the further information derivable from analysis of a single large set of related data, as compared to smaller sets with the same total amount of data, making correlations to be found for spotting business trends, avoid diseases, battle crime. Big data contains data sets with sizes beyond the ability of commonly used software tools to confine, manage, & process data within an acceptable beyond time. Big data "size" is a constantly moving objective, as of its ranging from a few dozen terabytes to several petabytes of data. Big data is a set of techniques and technologies that need new forms of integration to uncover large hidden values from large datasets that are diverse, complex, and of a massive range. Big data environment is used to acquire, organize and analyze the different types of data. We have number of complex network systems with cloud. In these complex network systems wireless sensor network and social network, data abnormality, and error develop into an annoying problem for the real network applications. So, researchers are attracted on network problem i.e., how to find data errors in complex network system for enhancing as well as debugging the network.

2. RELATED WORK

Chi Yang et. al describes a technique about a time efficient approach for detecting errors in big sensor data on cloud. It introduces that a big sensor data is prevalent where the data is generated with high volume and velocity such as both industry and scientific research applications. This is too complicate to process by using on-hand database management tools or traditional data processing applications. To address this problem cloud computing provides a flexible stack of massive computing, storage and software services with low cost as well as scalable manner. In existing, several techniques developed for processing device information on cloud, like sensor-cloud. However, these techniques don't offer economical hold on quick detection and locating of errors in huge device information sets. Xuyun Zhang et al introduces the theme of Proximity-Aware Local-Recoding Anonymization along with MapReduce for Scalable Big Data Privacy Preservation in Cloud describes, cloud computing offers promising scalable IT infrastructure to support various processing of a range of massive data applications in sectors like healthcare and business. Data sets like electronic health records in such applications often includes privacy-sensitive information, which brings about privacy concerns potentially if the information is distributed or shared to third-parties in cloud. A practical and widely-adopted technique for data privacy preservation is to anonymized data via generalization to satisfy a given privacy model. However, most existing privacy preserving approaches modified to small-scale data sets often fall short when encountering big data, due to their insufficiency or poor scalability. Huan Ke et al describe The MapReduce programming model simplifies large-scale processing on commodity cluster by working parallel map tasks and reduce tasks. Although many efforts have been made to improve the performance of MapReduce jobs, they ignore the network traffic created rated in the shuffle phase, which plays a critical role in performance enhancement. Historically, a hash function is used to partition intermediate data among reduce tasks, which, however, is not traffic-efficient since network topology and data size associated with each key are not taken into consideration. Bo dynasty et al place the construct on economical Feature Ranking strategies for High-throughput knowledge Analysis, here they had distinct efficient mining of high-throughput data has become one of the popular themes in the big data era. Existing biology connected feature ranking strategies in the main focus on applied math and annotation info. In this study, two efficient feature ranking methods are proposed. Multi-target regression and graph embedding are incorporated in an optimization framework, in addition to feature ranking is achieved by introducing structured sparsely norm. Unlike existing methods, the proposed methods have two advantages: (1) the feature subset simultaneously account for global margin information as well as locality various information. Consequently, both global and locality information are considered. (2) Features are selected by group rather than individually in the algorithm framework. Thus, the interactions between features are measured and the optimal feature subset can be guaranteed. In addition, this study presents a theoretical justification. Experiential experiments demonstrate the effectiveness and efficiency of the 2 algorithms compared with some state-of-the-art feature ranking methods throughout a collection of real-world organic occurrence data sets.

3. FRAMEWORK

A. Scale-Free Networks

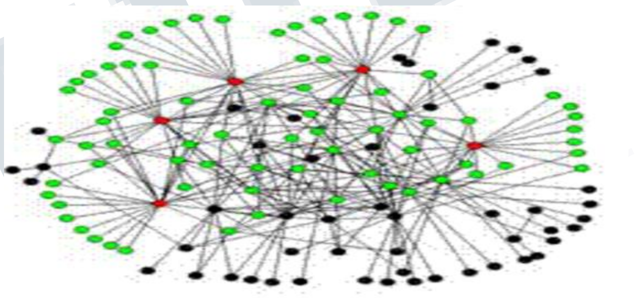


Fig. 1 Scale-free Network

Scale-free networks are shaped by random connections – the randomness of an ever-expanding Extremistan, that is. Erdös and Rényi set up a circle of nodes and then assigned a standard number of connections to each of them. Some nodes had more, several less, but overall the variation was constrained to a bell curve distribution. Yet as Barabási notes, this is a topographic or static approach to randomizing networks. It starts with a fixed number of nodes and wires them up in one go. He instead determined to explore a more active, semiotic and growth-based approach. His inspiration came from analyzing the connectivity patterns of the internet - the way that pages link to pages to form like-minded communities. As any surfer knows, there are several pockets of common interest on the web. One study of 100 million web pages by Jon Kleinberg claimed they might be parsed by subject substance into about 50,000 identifiable clusters. The groupings were of course loosely surrounded. So "randomly" there would be some surprising connections made across the breadth of the web. And within any community, several sites were always more prominent than others. These acted similar to hubs, with many other pages hanging off them, while additional sites were isolated and rarely visited. The pattern looked much similar to that for the world's air transport network. A small number of airports, such as Heathrow, Chicago and Frankfurt, are really big. Nearly everybody passes through them at some time even if they are heading on to other places. This type of network is multi-level a nested hierarchy. There are top-level airports that attach far and wide, then the regional airports that connect in a more nearby selective manner. It seems logical that an air transport or information network might be organized in this manner. But what Barabási and others managed to show was how such networks would spontaneously self-assemble during "randomness". That is, in a network along with scale in which the global and local levels are in free interaction and thus able to appear at their natural equilibrium balance then a scale-symmetric or fractal pattern of connectivity must emerge. It does not have to be considered in but self-organizes through its own devices.

B. Error Detection and Correction in Sensor Data

The essential idea of our proposed approach consists of using the properties of sensor data sources to detect and correct errors in the data. Sensor networks are planned to observe physical processes and typically designed to oversample the sensors, which results in considerable spatial and sequential correlations. The correlation provides built in redundancy within the data, which can be handled in unusual ways as part of the system design. One option could be to remove this redundancy close to the foundation by compression techniques, as is done in existing communication networks. Yet, while this improves usage of the communication channel, it also reduces the robustness of the data against errors as well as necessitates strong error coding methods that can add complexity to the end points. As an alternative, we propose to use this redundancy to correct errors from different sources. The approach is based on the difference among the correlation properties of the data & the sources of errors. For instance, the soft errors are completely uncoordinated, while the sensor data can have various types of correlation, depending upon the application, type of sensor, location of employment, etc. Since the correlation properties of the sensor data can differ depending on many factors, the success of this method depends on effectively identifying these properties. It also depends on how well the properties of the errors differentiate them from the data. There are two important benefits of being able to exploit the data properties for error correction. First, it makes the network design simpler and more efficient by addressing numerous types of errors together and removing the overheads of removing or adding any redundancy through compression or error coding. Second, it accesses all the processing required for error correction to be moved out of the sensor node. This can have a significant impact on the network architecture, since it lets the sensor nodes to become easier and use specialized nodes to perform error correction, possibly for multiple sensor nodes simultaneously.

C. MapReduce Model

The MapReduce programming model contains two functions at the API level: The map function transforms input data into (key, value) pairs, & the reduce function is applied to each list of values that correspond to the same key. This programming model abstracts away composite disseminated systems issues, thereby providing users with quick utilization of computing resources. To attain parallelism, the MapReduce system essentially implements "group data by key, then apply the reduce function to every group". This calculation model, referred to as MapReduce group by, permits parallelism because both the mining of (key, value) pairs & the application of the reduce function to each group can be performed in parallel on several nodes. The system code of MapReduce implements this computation model (and other functionality for instance scheduling, load balancing, as well as fault tolerance). The MapReduce program of an analytical query includes both the map and reduces functions compiled from the query (e.g., using a MapReduce-based query compiler) and the MapReduce system's code for parallelism.

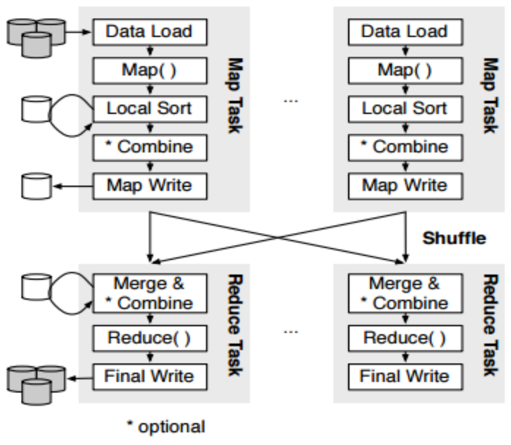


Fig.2 Big data implementation of MapReduce

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

D. Extensible Sensor Stream Processing's Declarative Sensor Data Cleaning:

Extensible Sensor stream Processing (ESP) is used to pervasive applications to build sensor data cleaning infrastructures. These several pervasive applications rely on data collection from physical sensor devices for instance, wireless sensor networks and RFID technology. In ESP sensor data cleaning includes five programmable phases:

Point
Smooth
Merge
Arbitrate
Virtualize

Point:
The main purpose of this phase is to filter individual values. ESP applies the point query to each sensor's readings.

Smooth:
In this phase, ESP utilizes the temporal granule described by the application to correct for missed readings as well as detecting errors in a solo sensor stream.
Here, both point and smooth phase's operations might be push down to capable sensor devices.

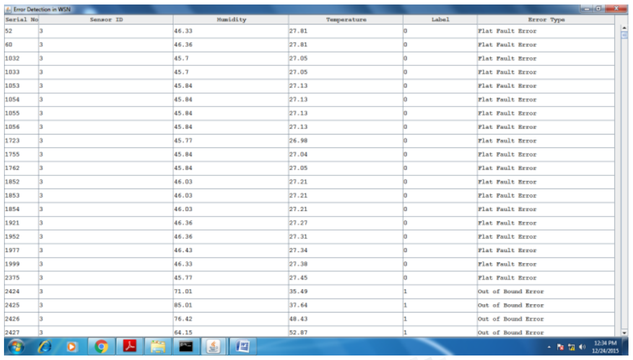
Merge:
Analogous to the sequential processing in the Smooth stage, Merge utilizes the application's spatial granule to correct for missed readings as well as remove errors spatially. At every time step, Merge processes input readings from a single type of device and groups the readings through the specified spatial granule using the GROUP BY clause.

Arbitrate:
This phase manages with conflicts, for instance duplicate readings among data streams from various spatial granules.

Virtualizes:
Eventually, few types of data cleaning utilize readings from across dissimilar types of sensors or stored data for enhanced data cleaning. The Virtualizes stage combines readings from different types of devices and different spatial granules to give a platform for such techniques.

4. EXPERIMENTAL RESULTS

In the experiments, we have to detect some types of errors in the dataset. For that we have to upload a dataset. After uploading the data set we have to apply the Map-Reduce function to partition the sensor data. After partition the sensor data we can detect the errors for instance, flat fault errors, out of bound errors, missing values errors.

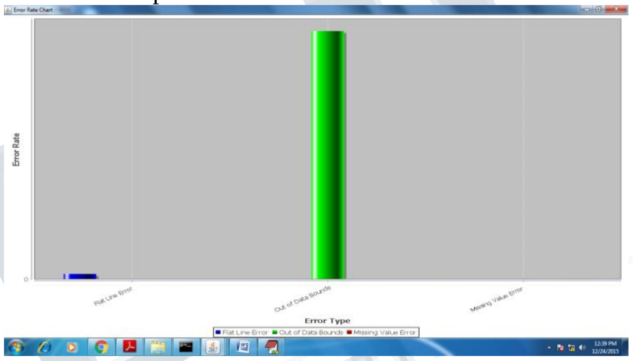


Error Detection in WSN

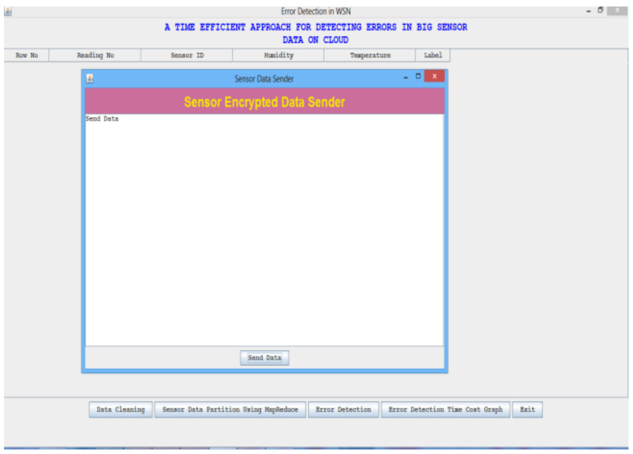
Serial No	Sensor ID	Humidity	Temperature	Label	Error Type
52	3	46.33	27.81	0	Flat Fault Error
60	3	46.36	27.81	0	Flat Fault Error
1032	3	45.7	27.05	0	Flat Fault Error
1033	3	45.7	27.05	0	Flat Fault Error
1053	3	45.84	27.13	0	Flat Fault Error
1054	3	45.84	27.13	0	Flat Fault Error
1055	3	45.84	27.13	0	Flat Fault Error
1056	3	45.84	27.13	0	Flat Fault Error
1723	3	45.77	26.98	0	Flat Fault Error
1755	3	45.84	27.04	0	Flat Fault Error
1762	3	45.84	27.05	0	Flat Fault Error
1852	3	46.03	27.21	0	Flat Fault Error
1853	3	46.03	27.21	0	Flat Fault Error
1854	3	46.03	27.21	0	Flat Fault Error
1921	3	46.36	27.27	0	Flat Fault Error
1952	3	46.36	27.31	0	Flat Fault Error
1977	3	46.43	27.34	0	Flat Fault Error
1999	3	46.33	27.38	0	Flat Fault Error
2375	3	45.77	27.45	0	Flat Fault Error
2424	3	71.01	35.49	1	Out of Bound Error
2425	3	85.01	37.64	1	Out of Bound Error
2426	3	76.42	48.43	1	Out of Bound Error
2427	3	64.15	52.87	1	Out of Bound Error

12:34 PM
12/24/2015

The above screen shows that detected different types of errors in the uploaded dataset.



In the above graph screen, we can observe that the time cost comparison among the different errors.
In our task we are including the expansion theme, as handling the dataset in a similar application is more weight to process .We are adding the sender application to process the dataset in scrambled arrangement which is more secure and handling the application in simple and secure way. We are utilizing the AES application to scramble the dataset.


Error Detection in WSN
A TIME EFFICIENT APPROACH FOR DETECTING ERRORS IN BIG SENSOR DATA ON CLOUD
Row No
Reading No
Sensor ID
Humidity
Temperature
Label
Sensor Data Sender
Sensor Encrypted Data Sender
Send Data
Send Data
Data Cleaning
Sensor Data Partition Using MapReduce
Error Detection
Error Detection Time Cost Graph
Exit

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

5. CONCLUSION

In this paper, we can conclude that, we proposed a time efficient approach for detecting errors and locating errors in big data sets on cloud through the scale-free networks. By using this proposed approach, we can reduce the time for detecting errors in big data sets. We can overcome the problems of data cleaning and error detection in the bid sensor cloud data by using proposed approach.

REFERENCES

[1] S. Sakr, A. Liu, D. Batista, and M. Alomari, "A Survey of Large Scale Data Management Approaches in Cloud Environments," IEEE Comm. Surveys & Tutorials, vol. 13, no. 3, pp. 311-336, Third Quarter 2011.

[2] B. Li, E. Mazur, Y. Diao, A. McGregor, and P. Shenoy, "A Platform for Scalable One-Pass Analytics Using MapReduce," Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD'11), pp. 985- 996, 2011.

[3] E. Elnahrawy and B. Nath, "Online Data Cleaning in Wireless Sensor Networks," Proc. First Int'l Conf. Embedded Networked Sensor Systems (ACM Sensys '03), pp. 294-295, 2003.

[4] M.H. Lee and Y.H. Choi, "Fault Detection of Wireless Sensor Networks," Computer Comm., vol. 31, no. 14, pp. 3469-3475, 2008.

[5] M.C. Vuranand and I.F. Akyildiz, "Error Control in Wireless Sensor Networks: A Cross Layer Analysis," IEEE Trans. Networking, vol. 17, no. 4, pp. 1186-1199, Aug. 2009.

[6] R. Albert, H. Jeong, and A. L. Barabasi, "Error and Attack Tolerance of ComplexNetworks," Nature, vol. 406, pp. 378-382, July 2000.

[7] D.J. Wang, X. Shi, D.A. Mcfarland, and J. Leskovec, "Measurement Error in Network Data: A Re-Classification," Social Networks, vol. 34, no. 4, pp. 396-409, Oct. 2012.

[8] D. Xiong, M. Zhang, and H. Li, "Error Detection for Statistical Machine Translation Using Linguistic Features," Proc. 48th Ann. Meeting of the Association for Computational Linguistics (ACL'10), pp. 604-611, 2010.

[9] S. Mukhopadhyay, D. Panigrahi, and S. Dey, "Model Based Error Correction for Wireless Sensor Networks," IEEE Trans. Mobile Computing, vol. 8, no. 4, pp. 528-543, Sept. 2008.

[10] K. Ni, N. Ramanathan, M.N.H. Chehade, L. Balzano, S. Nair, S. Zahedi, G. Pottie, M. Hansen, M. Srivastava, and E. Kohler, "Sensor Network Data Fault Types," ACM Trans. Sensor Networks, vol. 5, no. 3, article 25, May 2009.

[11] 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," Int'l J. Distributed Sensor Networks, vol. 2013, pp. 1-18, 2013.