Journal of Advanced Zoology
ISSN: 0253-7214
Volume 44 Issue S-8 Year 2023 Page 463-470

Memory Management In Real-Time Mining Of Massive Complex Data Streams

Kavitha N[1*], Dr.Y. Kalpana[2], Dr.Kumar V[3]

[1]Department of Information Technology, Vels Institute of Science Technology and Advanced Studies, Chennai, Tamilnadu, India, kavitanithyanandam@gmail.com*
[2]Department of Information Technology, Vels Institute of Science Technology and Advanced Studies, Chennai, Tamilnadu, India, ykalpanaravi@gmail.com
[3]Former Professor and Head, Agricultural, Engineering Department, ACRI, Madurai, Tamil Nadu, India, vskumaran1955@gmail.com

***[*]Corresponding Author:** Kavitha N*
[*]Department of Information Technology, Vels Institute of Science Technology and Advanced Studies, Chennai, Tamilnadu, India, kavitanithyanandam@gmail.com

Abstract

The terms "real-time mining and streaming of data" have become gained immense popularity in the data field where they have access to the fastest and the latest data on a real-time basis. Real-Time-Mining attempts to develop a real-time framework to minimize adverse environmental impact and increase resource efficiency. The real-time analysis deals with a huge rate of change in data which needs to be processed and updated frequently and rapidly. Data Mining encompasses a multi-disciplinary field. This combines several domains such as artificial intelligence (AI), statistics, machine learning, database technology, etc. The key objective of data mining is to explain the past and predict the future. This is achieved by exploring and analyzing a huge amount of data almost on a real-time basis from diverse datasets and sources. This process can be termed Knowledge Discovery. Data Mining endeavors to store the data in the local data set, hosted by local computers that are connected to the computer networks. In the real world, data has become large and almost unmanageable with several data streams. Extraction of numerous knowledge structures from continuous and rapid data records is called data stream mining. A data stream includes an ordered sequence of several instances. The latter can be read only once or a few times in many data stream mining applications by employing the available computing and storage capabilities in the information technology world. Though the technology comes to real-time distributed mining of complex data streams, ample research has already been conducted on decreasing computation cost, ensuring enhanced data privacy at the distributed sites, and optimal deployment of limited assets. The key characteristics of mining complex data streams include huge volume of continuous incoming infinite data; the nature of the data is fast-changing, necessitating a fast real-time response. The data become multi-dimensional in nature. Since the data set is complex, some of the

	challenges to be addressed are unbounded memory requirements. The current paper analyses how effectively memory can be managed in real-time data streams.
CC License CC-BY-NC-SA 4.0	***Keywords: data mining, data stream, data cleaning, cron, voltDB, multiple databases, IEP, FFM cap***

INTRODUCTION

In recent years, data inflow has been abundant, and the output of the data is very minimal. The reason behind the minimum amount of research on the data is because of the memory and storage of the data. In data streams, data inflows continuously, so analysing the data inflow also becomes challenging. The first challenge we encounter is the safe storage of the data, and then its analysis in a cost-effective manner becomes important. If the data are properly and safely stored and are made available in the future for analysis, they can be effectively analysed. This can make the predictions more effective and accurate. The current study evaluates different methodologies of storing the data in multiple databases and gives an overview of how to extract the relevant information from the stored data and how to effectively use the data to predict a better output.

MINING OF DATA STREAMS

DataStream mining, also called stream learning, is the extraction of required knowledge from continuous and rapid records. Different characteristics of data stream mining are a continuous stream of data, concept drifting, and data volatility. Any data stream can enter the system and provide elements at its schedule. The real-time data do not have the same data rate or data type. The time between one stream and the other may not be uniform. Several streams of data can be archived for the minimum time and examined under special circumstances for a defined retrieval process. Some of the challenges faced in stream mining are unbounded memory management, approximate query answering, high-quality output, etc. Risk management, e-Commerce, Network monitoring, Fraud detection, etc. are some of the key data mining applications.

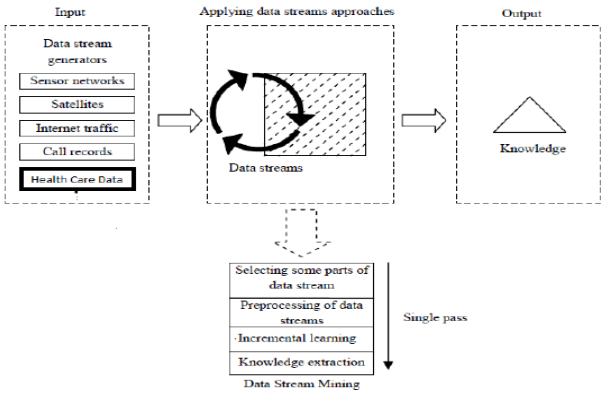

Fig. 1. The general process of data streams mining

ARCHITECTURE OF THE RESEARCH

Real-time stock market data is collected. Collected data can be incorrect, poorly formatted, or irrelevant, and unclean. Data cleaning focuses on the updation, as well as correction and consolidation of the collected data. This is done with a view to ensuring a maximum effective system. Once the data are cleaned, knowledge extraction is performed from the base data. Once the information is extracted completely, data is relayed to different databases for further analysis. Here the primary data will be in a single database, that participates in the model prediction. The other information will be pushed to multiple databases where the analysis will be done as and when required.

Fig. 3. The architecture of Data Cleaning

DATA CLEANING

Data cleaning fixes, removing wrong, corrupted, duplicate, and incomplete records from the data set. Incorrect data outcomes will be unreliable, and the outcomes will be incorrect. As datasets vary, there is not a uniform and standardized way to ensure data cleaning. After completing the data cleaning, the data transformation is done. It relates to converting the data from one format to the other format or structure for easy process, evaluation, and extracting the relevant latest information for the application. Cleaning data will ensure error-free data, data quality, accuracy, and efficiency and maintain data consistency.

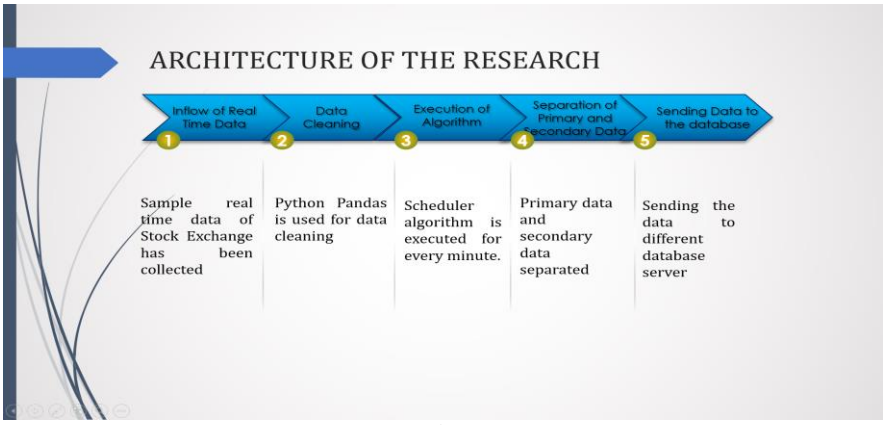

Fig. 2. The architecture of the research

STEPS FOR DATA CLEANING

Different steps adopted for cleaning the dataset are given below:

1: Rename the column header using the common column name used in the database.
2: Look for the statistical characteristic of the dataset.
3: Drop the full NULL rows.
4: Replace the null, N/A and Nan values with 0.
5: Cast the column in the dataset (i.e.) data transformation, as per the table structure.
6: Check for the statistical analysis of the data.

SAMPLE DATASET

In this study, the sample data set was collected from the National stock exchange of India Ltd. The dataset contains the base data such as symbol, previous closure value, IEP (Indicative Equilibrium Price), Total quantity, FFM Cap (Free-Float Methodology Capitalization), NM 52W H, and NM 52W L. Other data are determined from the base data, and the calculated data will participate in the model.

SYMBOL	PREV. CLOSE	IEP PRICE	FINAL QUANTITY	FFM CAP	NM 52W H	NM 52W L
ITC	241.5	243.5	369365	21,10,53,59,76,239.88	242.45	163.35
INDUSINDBK	1,137.55	1,145.00	14753	6,86,62,16,39,853.40	1,186.50	485
MARUTI	6,777.00	6,820.00	2358	9,00,76,64,89,312.80	8,329.00	6,270.15
ASIANPAINT	3,309.20	3,328.50	6725	14,91,86,33,43,533.96	3,394.60	1,907.75
TATASTEEL	1,292.20	1,299.40	39975	10,09,93,84,82,565.74	1,534.50	342.75
HINDUNILVR	2,812.45	2,827.90	5939	25,11,07,84,28,424.58	2,859.30	2,000.05
COALINDIA	156.4	157.25	32513	3,27,70,92,41,516.55	165	109.55
TITAN	2,085.85	2,096.80	2628	8,70,34,07,18,062.92	2,150.00	1,076.00
BAJFINANCE	7,813.85	7,850.00	8380	20,75,21,40,95,978.43	7,847.10	3,006.90
ICICIBANK	711.85	715	95842	49,29,30,49,81,864.45	734.8	333.75
JSWSTEEL	668.95	671.8	11427	6,46,79,98,45,335.20	776.5	257.55
HINDALCO	451	452.9	43359	6,58,67,89,13,528.35	488	154.4
NTPC	124.1	124.6	31420	5,89,64,45,70,942.41	125.7	78.1
TATAMOTORS	301.95	303.15	50593	5,41,38,61,42,006.55	360.75	122.15
TATACONSUM	858.9	862	2457	5,14,48,84,99,208.78	889	458.95
KOTAKBANK	2,010.95	2,016.50	10682	29,49,99,75,52,285.55	2,049.00	1,230.60
WIPRO	667.6	669	11170	9,87,64,10,64,440.03	690	302.45
EICHERMOT	2,854.40	2,860.00	1749	3,97,98,35,54,995.97	3,037.00	2,015.20
BPCL	413.8	414.5	13489	3,85,98,38,17,750.90	503	325
BRITANNIA	4,075.55	4,082.00	757	4,81,01,86,84,043.77	4,153.00	3,317.30
TECHM	1,461.70	1,464.00	4382	9,06,47,73,01,921.09	1,493.15	747.55
RELIANCE	2,404.70	2,408.00	37120	77,74,67,40,24,038.64	2,480.00	1,830.00
SHREECEM	30,059.90	30,100.00	43	4,01,29,59,60,417.92	32,048.00	18,183.55
ULTRACEMCO	7,592.70	7,599.90	364	8,76,66,66,50,931.84	8,073.30	3,753.90
BAJAJ-AUTO	3,728.80	3,732.00	542	4,85,54,62,84,879.20	4,361.40	2,822.45
SUNPHARMA	764.5	765	4599	8,25,43,12,13,054.25	804.35	452.25
M&M	738.45	738.9	3784	7,06,88,73,61,269.94	952.05	567.5
ADANIPORTS	749.6	750	7421	5,50,97,89,63,216.42	901	312.1
BHARTIARTL	726.65	727	21875	17,16,03,60,94,145.65	740.75	394

Fig. 4. Sample Data Set

ALGORITHM FOR EXTRACTING INFORMATION

Step 1: Start
Step 2: Declare variables for Prev close, IEP price and Quantity
Step 3: Read values for Prev close, IEP price and Quantity
Step 4: Formula for Change value = IEP price - Prev close
Step 5: Formula for % Change value = (Change value / Prev close) * 100
Step 6: Final price = IEP price
Step 7: Value = IEP price * Quantity
Step 8: Data is pushed to multiple databases.
Step 9: End

Previous closure, IEP, and the quantity will be the base value for calculation. Change of value is the difference between IEP price and the previous closure. The percentage of change of value is the percentage of change value and the previous closure. The final price remains the same as the IEP price. The quantity which shows the stock availability of the symbol, respectively, is taken into account. The value of each symbol is derived from the IEP price and quantity. A calculation algorithm is executed between the stream processing engine and the database.

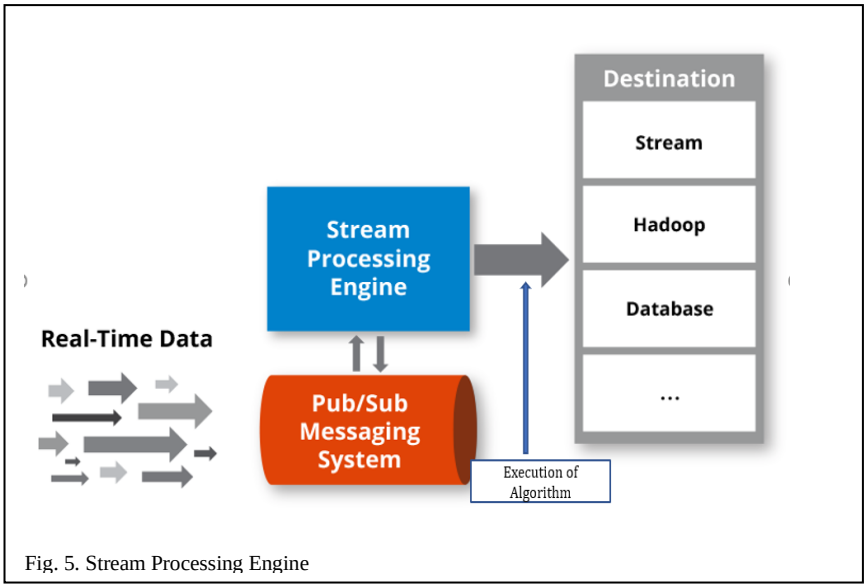

Fig. 5. Stream Processing Engine

CRON SCHEDULER

Cron Job is a Cron command-line utility used in a Unix-like operating system in order to maintain the scheduled jobs to run periodically at fixed times and intervals. This system automates both system maintenance and administration. The calculation algorithm script is executed every minute in the Cron scheduler for the calculating value. For each instance of data, inflow values will be calculated automatically. Once the calculation is done, another script for pushing the data to multiple databases will be executed. For the execution of the scripts in python, the following inbuilt algorithm is used by the Cron as follows.

1. First, look for a file named. Cron tab in the home directories of all account holders.
2. For each Cron tab file found, the next time in the future is determined so that each command can run properly.
3. Thereafter, these commands are placed on the Franta–Maly event list with their corresponding time and their "five fields" time specifier.
4. Lastly, the main loop is examined:

- Examine the task entry at the head of the queue, and compute the time required in the future to run it.
- Sleep for that period.
- On awakening and after verifying the correct time, execute the task. Task is executed at the head of the queue with creating user's privileges.
- Thereafter, the next time in the future is determined to run this command and place that on the event list at the required time value.

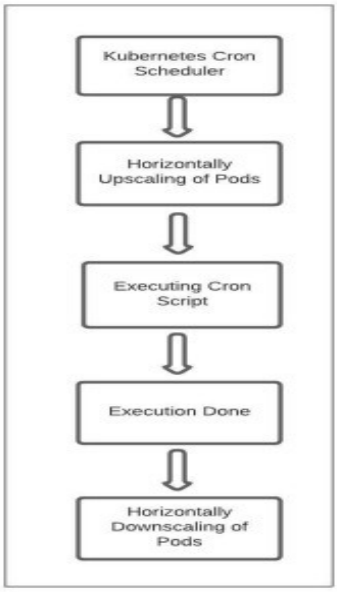

Fig. 6. The architecture of Cron Scheduler

CRON IMPLEMENTATION

The process involves running the script on the server according to the schedule. A command-line terminal is created on our local machine, and the job of running and executing the script will be taken care of by Kubernetes Cluster. Deployment of the script is done by creating the image through CI/CD pipeline using Docker and Jenkins, where a docker image is created and successively pushed into ECR. The YAML file is created, and Kubernetes will allocate resources like environmental variables and scheduler configuration. If resources are ready, the scheduler is started for the execution.

Once the Cron job is done, data pushing will be initiated along with the time stamp. The database used for storing and post processing is VoltDB.

```
FROM openjdk:8-jdk-alpine
COPY --chown=appuser:appuser target /home/appuser/target
COPY --chown=appuser:appuser start.sh /home/appuser/start.sh
WORKDIR /home/appuser
CMD ["sh", "start.sh"]
```

Fig. 7. Docker File Configuration

VOLTDB

VoltDB is an in-memory database and an ACID-compliant relational database management system (RDBMS). It employs share-nothing architecture, supporting the SQL access form. Then we optimize the VoltDB database for a specific application. This is achieved by partitioning the database tables to create the distributed database. VoltDB strikes a delicate balance for ensuring maximum performance requirements. Using this database, data is pushed to the different databases in the pre-defined table structure. Basic Hardware and Software requirements for using VoltDB are shown below:

Operating System	VoltDB requires a 64-bit Linux-based operating system. Kits are built and qualified on the following platforms: • CentOS version 7.0 and later, or version 8.0 and later • Red Hat (RHEL) version 7.0 and later, or version 8.0 and later • Ubuntu versions 18.04 and 20.04 • Macintosh OS X 10.9 and later (for development only)
CPU	• Dual core[1] x86_64 processor • 64 bit • 1.6 GHz
Memory	4 Gbytes[2]
Java[3]	VoltDB Server: Java 8, 11 or 17 Java and JDBC Client: Java 8, 11, or 17
Required Software	Time synchronization service, such as NTP or chrony[4] Python 3.6 or later
Recommended Software	Eclipse 3.x (or other Java IDE)

Fig. 8. Basic requirement for VoltDB

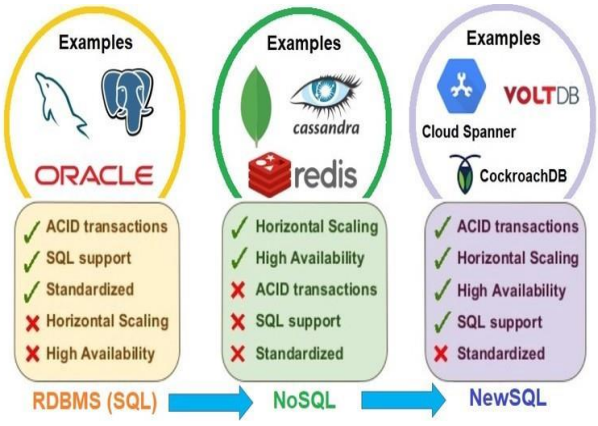

Fig. 9. Comparison of Databases

CONCLUSION

Overall, the data are cleaned, and the required values are calculated using the algorithm. Once the values are calculated, data is pushed to multiple databases. Here the primary data are in one single database, and the rest of the values will be pushed to a different database with the common column symbol. By achieving this, memory will be managed, and the unused columns will be on different servers. Unused data can be analysed in the scheduled intervals and can be discarded after the analysis. By discarding the analysed values, memory will be free, and it optimizes the overall system performance; the load on the machine will be reduced. The future scope of the research will be on model prediction and evaluating the overall model performance using machine learning.

ACKNOWLEDGMENT

We sincerely extend our sincere thanks to Dr. Y. Kalpana, Professor VISTAS for guiding me in my research with innovative ideas and advanced techniques. Also providing support as and when required. I would like to thank my professor Dr. Kumar V who seeded the idea of research in my career. I like to thank Dr. G. Suseendran (Late) for providing me with the topic of research and giving me the opportunity to work with VISTAS.

REFERENCES

1. Bifet and R. Kirkby, Data Stream Mining A Practical Approach.
2. S. K. Sen and B. K. Ratha "A Comprehensive Study on Distributed Data Mining and Learning Algorithms," Xi, J. B. Ni, "Deploying Mobile Agents in Distributed Data Mining," PAKDD 2007 Workshops, pp. 322–331, 2007.
3. S. Bailey, R. Grossman, H. Sivakumar, and A. Turinsky, "Papyrus: A System for Data Mining over Local and Wide Area Clusters and Super-Clusters".
4. V. Sawant and K. Shah, "A review of Distributed Data Mining using agents", International Journal of Advanced Technology & Engineering Research (IJATER), vol. 3, no. 5, pp. 27-33, 2013.
5. S. Kumar, P. N. Santosh Kumar, and C. Venugopal, "An Apriori Algorithm in Distributed Data Mining System", Global Journal of Computer Science and Technology Software & Data Engineering, vol. 13, no. 12, 2013.
6. K. Das, K. Bhaduri, and H. Kargupta, "A local asynchronous distributed privacy preserving feature selection algorithm for large peer-to peer networks", J. Knowledge and Information Systems, vol. 24(3), pp. 341-367, Sept. 2010.
7. R. Vilalta, C. Giraud-Carrier, P. Brazdil, and C. Soares, "Using Meta-Learning to Support Data Mining," International Journal of Computer Science & Applications, vol. 1, no. 1, pp. 31-45, 2004.
8. S. C. Frank, Y. H. Tseng, and Y. H, Min, "Toward boosting distributed association rule mining by data de-clustering," Journal of Information Sciences, vol. 180, no. 22, pp. 4263-4289, Nov. 2010.
9. G. S. Bhamra, A. K. Verma, and R. B. Patel, "Agent Enriched Distributed Association Rules Mining," ADMI 2011, pp. 30–45, 2012.
10. J. Costa da Silva and M. Klusch, "Inferences in Distributed Data Mining", Engineering Applications of Artificial Intelligence, vol. 19, pp. 363 -369, 2006.
11. M. A. Naeem, "A robust join operator to process streaming data in real time data warehousing," in Eighth International Conference on Digital Information Management (ICDIM 2013), pp. 119–124, IEEE, 2013.
12. H. Isah, T. Abughofa, S. Mahfuz, D. Ajerla, F. Zulkernine, and S. Khan, "A survey of distributed data stream processing frameworks," IEEE Access, vol. 7, pp. 154300–154316, 2019.
13. M. A. Naeem, G. Dobbie, and G. Weber, "Efficient processing of streaming updates with archived master data in near-real-time data warehousing," Knowledge and information systems, vol. 40, no. 3, pp. 615–637, 2014.
14. M. Babar and F. Arif, "Real-time data processing scheme using big data analytics in internet of things based smart transportation environment," J. Ambient Intelligence and Humanized Computing, vol. 10, no. 10, pp. 4167–4177, 2019.
15. N. Biswas, A. Sarkar, and K. C. Mondal, "Efficient incremental loading in etl processing for real-time data integration," Innovations in Systems and Software Engineering, pp. 1–9, 2019.

16. M. A. Naeem, G. Dobbie, I. S. Bajwa, and G. Weber, “Resource optimization for processing of stream data in data warehouse environment,” in Proceedings of the International Conference on Advances in Computing, Communications and Informatics, pp. 62–68, ACM, 2012.
17. R. Mukherjee and P. Kar, “A comparative review of data warehousing etl tools with new trends and industry insight,” in 2017 IEEE 7th International Advance Computing Conference (IACC), pp. 943–948, IEEE, 2017.
18. H. Bouali, J. Akaichi, and A. Gaaloul, “Real-time data warehouse loading methodology and architecture: a healthcare use case,” Int. J. Data Analysis Techniques and Strategies, vol. 11, no. 4, pp. 310–327, 2019.
19. R. Duan, R. Prodan, and T. Fahringer, "Short Paper: Data Mining-based Fault Prediction and Detection on the Grid," High Performance Distributed Computing, 15th IEEE International Conference on High Performance Distributed Computing, vol., no., pp.305-308, 2006
20. N. Khayat, Semantic Instrumentation and Measurement of Data Mining Algorithms, Technical Report on R&D 2, Hochschule Bonn-Rhein-Sieg, 2009.
21. S. Datta, K. Bhaduri, C. Giannella, R. Wolff, and H. Kargupta. Distributed data mining in peer-to-peer networks. Internet Computing,IEEE, vol. 10, no. pp. 18–26, 2006.
22. N. Khan, I. Yaqoob, I. A. Hashem, Z. Inayat, W. K. Ali, M. Alam, M. Shiraz, and A. Gani, “Big data: survey, technologies, opportunities, and challenges,” Scientific World J, Article ID 712826, 2014
23. M. Steen, G. Pierre, and S. Voulgaris, “Challenges in very large distributed systems,” J Internet Serv Appl, vol. 23, no. 1, pp. 59–66. G. Tsoumakas, and I. Vlahavas, Distributed data mining. In: Encyclopaedia of Data Warehousing and Mining, IGI Global, Hershey, PA, USA, 2009, 709–715.
24. S. Cong, J. Han, J. Hoeflinger, and D. Padua, A sampling-based framework for parallel data mining. In: Proc. of the ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, Chicago, Illinois, USA, 2005, 255–265.
25. P. Luo, K. Lü Z. Shi, and Q. He, “Distributed data mining in grid computing environments,” Future Gener Comput Syst,” vol. 23, no. 1, pp. 84–91, 2007.
26. M. Last, “Online classification of nonstationary data streams,” Intelligent Data Analysis, vol. 6, no. 2, pp. 129-147, 2002.
27. Shearer, “The CRISP-DM model: the new blueprint for data mining,” J. Data Warehousing, vol. 5, no. 4, pp. 4-15, 2000.
28. Aggrawal, Data Streams: Models and Algorithms, Springer, 2000
29. L. O'Callaghan, N. Mishra, A. Meyerson, S. Guha, and R. Motwani, “Streaming-data algorithms for high-quality clustering,” in Proc. 2003 IEEE International Conference on Data Engineering.
30. M. M. Gaber, S. Krishnaswamy, and A. Zaslavsky, On-board mining of data streams in sensor networks advanced, Methods of Knowledge Discovery from Complex Data, Springer, pp.307-335, 2006
31. S. Muthukrishnan, Data streams: algorithms and applications, Proceedings of the fourteenth annual ACM-SIAM symposium on discrete algorithms, 2003