ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2014

Visual Exploration of Amnesic Time Series Data Streams

Kaushal Chauhan[1,] Mukta Takalikar[2]

Research Scholar, Department of Computer Engineering, Pune Institute of Computer Technology, Pune, India[1]

Associate Professor, Department of Computer Engineering, Pune Institute of Computer Technology, Pune, India[2]

ABSTRACT: Time Series data is a time oriented data, where each data item refers to a specific point measured typically at successive instances in time space. Streaming data is real time, potentially massive, rapid sequence of data information arriving continuously in ordered sequence of items. Various researches have been carried out that focused on representations which are processed in batch mode and visualize each value with almost equal dependability. In many domains recent information is more useful than older information. We call such incoming data as amnesic as it consists of greater value for data analysis. The dissertation proposed a novel system to monitor streaming amnesic time series data, handle data streams by using sliding window and memory management methods, summarizing the amnesic data with the help of weighted moving average algorithm. Final phase includes visualizing amnesic and summarized data streams in the form of dynamic line chart visualization and generating the reports of summarized data as snapshots, which eventually facilitates analysts to recognize various patterns underlying streaming time series data.

KEYWORDS: Data Streams, Summarization, Amnesic, Time Series Data, Visualization.

I. Introduction

Time-series data is vitally used in science, engineering and business. Visualization helps individuals interpret information by exploiting human perception and to scale the psychological feature of visuals. Statistical graphics, most notably line charts of time-value pairs, are heavily used for inspecting individual or tiny sets of time series [1]. However, understanding massive collections of time series data remains tough. We tend to elite large-scale system management as a site wherever individuals have to be compelled to perceive large sets of time-series information at multiple levels of detail and with reference to often ever-changing groupings [2] [3].

Data warehouses for managed hosting services will store details regarding tens of thousands of physical and virtual servers. For every system, parameters like mainframe load and memory usage are often logged [4]. This information could also be archived for multiple years. System management employees should be ready to question elaborated information to attend to the wants of individual customers, whereas maintaining awareness of the managed environment's global state [5].

II. Related work

Many infinite stream algorithms do not have obvious counterparts in the sliding window model. For instance, while computing the maximum value in an infinite stream is trivial, doing so in a sliding window of size N requires Ω (N) space; consider a sequence of non-increasing values, in which the maximum item is always expired when the window moves forward. Thus, the fundamental problem is that as new items arrive and old items must be instantaneously removed for further processing [6].

Generally the time oriented data is the data that are linked to time. In the other sense we can assume that it is data generated with the time stamp. Certainly, this general description isn't sufficient once users need to select or developers need to develop applicable visualization ways. A vital demand for achieving communicative and effective visualization is to think about the characteristics of the information to be given, which, in our case, are significantly associated with the dimension of time. Various methods have been derived with respect to formulation on time in several areas of engineering, as well as AI, data processing, simulation, modelling, databases, and more [7].There exist many techniques for message type extraction and event identification. Most of these techniques make two to three scan over log file to generate message types and one pass to identify events using this message types[8].

Using regular expression for each distinct token is one of these techniques. This has some disadvantages such as it requires full knowledge of system and is suited for log file containing few distinct events. As our focus is on log analysis this technique is useful as it is simple and take less time compared to other techniques [9].

Traditional databases are utilized in applications that need persistent data storage and sophisticated querying. Usually, information consists of a collection of objects, with insertions, updates, and deletions occurring less frequently than queries. Queries are executed once entered and therefore the answer reflects this state of the database. Since previous few years had observed the emergence of applications that don't work this knowledge model and querying paradigm. Instead, data naturally happens within the variety of a sequence streaming values; examples embrace sensor data, net traffic, money tickers, on-line auctions, and transaction logs like internet usage logs and telephone records [10] [11].

In addition to windowed sampling [14], a possible solution to computing sliding window queries in sub linear space is to divide the window into small portions called as basic windows [12] and only store a synopsis and a timestamp for each portion. When the timestamp of the oldest basic window expires, its synopsis is removed, a fresh window is added to the front, and the aggregate is incrementally recomputed. This method may be used to compute correlations between streams [12], Find frequently appearing items [15], and compute various aggregates [4] [16] are some of the operations performed on data streams.

However, some window statistics may not be incrementally computable from a set of synopses. The symmetric hash join and an analogous symmetric nested loop join may be extended to operate over two [19] or more sliding windows by periodically scanning the hash tables (or whole windows) and removing stale items. Interesting trade-offs appear in that large hash tables are expensive to maintain if tuple expiration is performed too frequently [20]. Time series data can analyzed by applying various approximation methods, some of clustering techniques described in [29] gives better approaches for analysis and these streams after processing can be visualized by various visual methods to represented the time oriented data [30]. The appropriate visualization methods can be applied which are reviewed on the basis of various aspects such as visualization methods, types of variable, mapping techniques, dimensionality with reference to varying attributes of data streams [32] [33]. The dynamic and interactive visual have to be taken in consideration for user’s easy access and analytical purpose. Various visualization techniques for time series data streams present till date are briefly described and summarized in [34].

III. Mathematical Modeling

System is defined as:

$S = \{D, H, S_{sum}, V, F| \Phi_s\}$ where

D – Data Streams

S_{sum} – Stream Summarization

H – Hierarchies set for data streams

V - Views for data display

F - Set of functions and

Φ - Constraints for the system.

System components are as explained below:

$D = \{D_{inp}, D_p, D_\delta, D_v, D_{seg}, D_{sum}, D_g\}$ where

1.$D_{inp} = \{D_1, D_2 \ . \ . \ .D_n\}$ is input financial time series data in relational format that needs to be visually explored where

$Di = \{A_a, A_t, A_m\}$, individual record of data stream stored in buffer

i. $A_a = \{ T_{id}, T_{id,} D_{id}, S_d, M_a \}$, Address component in the data record

T_{id} = Time ID

T_{id} = Series ID
D_{id} = Data Point ID
S_d = Series Data
M_a = Multivariate attributes
ii. A_t = Time component in the data record
iii. A_m = Analysis attributes

2. D_p = Pre computed data stream.
3. D_{seg} = Segmentation information for data stream.
4. D_δ = Mean Deviation for each time series.
5. D_{sum} = summarized time series data.
6. D_g = grouped time series data.

$S_{sum} = \{W_{ma}, N_t\}$ represents the summarization method where
W_{ma} (m) = function to calculate weighted moving average of data stream
N_t = Forecasted value of data point

1. $W_{ma} = \mu + A(n) - \O_1.A(n-1) - \O_2.A(n-2) - \ldots - \O_m.A(n-m)$
   i. μ = Mean of Time Series
   ii. A(n) = White Noise
   iii. $\O_1, \O_2 \ldots \O_m$ = Parameters of Model
   iv. m = Order of Moving Average Model
2. $N_{t+1} = wt_1(V_t) + wt_2(V_{t-1}) + \text{---} + wt_n(V_{t-m})$
   i. N = Forecasted value
   ii. t = Total number of data point in window
   iii. wt = weight of moving order
   (Where, $wt_1 + wt_2 + \text{---}\ wt_n = 1$)
   iv. V = Value of Stream

$V = \{M, \zeta, \tau, D_v, F_d\}$ represent view for displaying data where
1.$M = \{S, \beta, M_{ir}, M_t\}$ is a dynamic chart where
   i. $S = \{S_1, S_2 \ldots S_m\}$ set of scale where
   ii. S_i = {Xaxis, Yaxis}
   iii. Xaxis = scale to set on X axis
   iv. Yaxis = scale to set on Y axis
   v. β represent time interval
   vi. M_{ir} = value for chart region e.g. sec, hour, month etc.
   vii. M_t = Type of chart such as line chart, area chart, pie chart etc.
2.$\zeta = \{X, Y, \psi\}$ is a chart view where
   i. X = X axis variable
   ii. $Y = \{Y_1, Y_2, \ldots, Y_x\}$ Y axis variables
   iii. ψ = Statistical data associated with the chart such that $\psi \subset D_v$
3. τ = Series Text display
4. D_v = Streaming data associated with the view.
   Please note that the Data D_v is common for the views, so data is retrieved once and used for computation related to all view components. This also helps in implementing linked views. Linked view means highlighting a particular record in one of the View components would highlight similar record in another view component.
5. F_d = Filter control for data selection

$H = \{H_1, H_2, \ldots, H_k\}$ Hierarchies present in the data streams based on the classification with respect to data attributes where
$H_i = \{H_l, A_l\}$ where

i. H_l = Level in the hierarchy
ii. A_l = Attribute for hierarchy level that corresponds to one of the address components in the data

F = Set of functions consisting of following:

1. F_g (A_a) -> L_i, a function to convert address into charting view.
2. F_{seg} (D_{inp}, k) ->D_{seg}, a function for segmenting the data.
   Data segments are identified for each analysis attribute based on minimum and maximum values of that attribute.
3. F_p (D_{inp}) ->D_p, a function for pre computing statistics. For each data point, cumulative mean over selected year range is calculated as:
   $A_{mmean}[t_i] = (\sum_{t_{Start}}^{i} A_m\,[t_i])/(t_i - t_{Start})$
4. F_δ(D_{inp}) ->D_δ, a function for calculating mean deviations for each time series. Absolute Mean deviation for time series for each series form defined window is calculated as:
   i. For each data point in the stream, Value of A_m for t_i is mean value of that analysis attribute for that stream window over the entire time range. Let us denote this as W.$A_{m\delta}$.
   ii. Now we find out the average of all the time series as
   $L_{mean} = (\sum_{i=0}^{n} L.\,A_{m\delta})/n$
   iii. Mean deviation for each window of time series is
   $L_{i\delta} = ((|\,L_{mean} - L.A_{m\delta}|)/\,L_{mean})$*Window Size
5. F_h-> (D_{inp}) -> H, a function to formulate hierarchy from the input data stream.
6. F_v(D) -> V, a function to display view.
7. F_s (D,A_t) -> V, a function Show simulation over a selected time range.
8. F_d(M) -> V, a function to handle data point plotting on chart.
9. F_{accum}(L) -> τ, a function to handle data point accumulation on chart.
10. F_{frame}(F) -> V, a function to handle frames of view.
11. F_{grp} (D_p) ->D_g, a function to group the time series data streams on view.
    i. Uses following distance calculation and Centroid calculation formulae.
    ii. The centroid of a non-self-intersecting closed polygon defined by n vertices (x_0, y_0), (x_1,y_1), ..., (x_{n-1},y_{n-1}) is the point (C_x, C_y), where
    $$C_x = \frac{1}{6A}\sum_{i=0}^{n-1}(x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i)$$
    $$C_y = \frac{1}{6A}\sum_{i=0}^{n-1}(y_i + y_{i+1})(x_i y_{i+1} - x_{i+1} y_i)$$
    and$A$ is the polygon's signed area,
    $$A = \frac{1}{2}\sum_{i=0}^{n-1}(x_i y_{i+1} - x_{i+1} y_i)$$
12. F_f(D_p, k, n) ->D_f, a function to calculate forecasted data point for given set of data streams.
    k is number of requested forecasts, n is number of intervals to consider for forecasting.

Success:

- Summarized Time Series Data Streams.
- Interactive & Graphical Streaming Visualization in the form of dynamic Charts.

Failure:

- Errors in Time Series data streams beyond Threshold Level.
- Graphical memory buffer over flow
- Dynamic visual lags or slow in responsive.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2014

IV. SIMULATION RESULTS

This section provides the performance and accuracy results of Time Series Visualization System.

1.Display Time

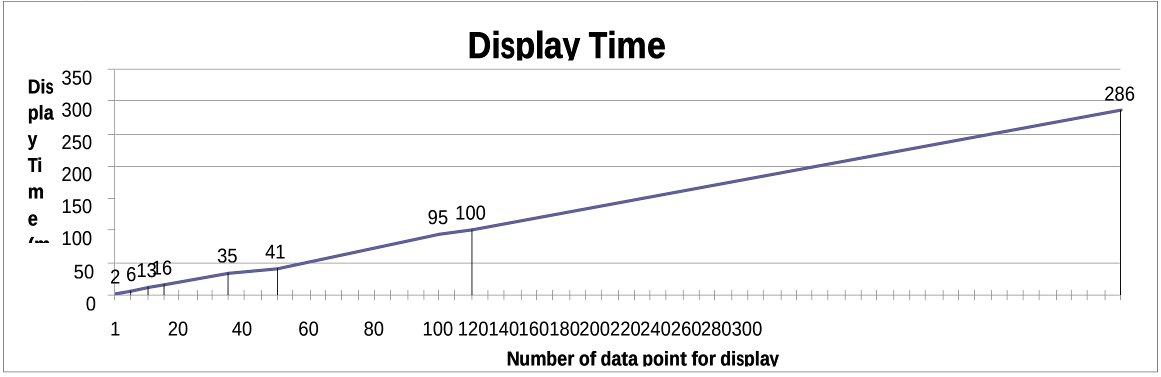


Fig. 1. Display time chart

From the figure 1, we conclude that:

- To show a data point on chart, average display time is 1.2 ms.
- As the number of data point increase, the display time per data point stabilizes and does not increase.

2.Processing Time

No of Streams	Weight of Window	No of Analysis Variables	Time for pre-processing (ms)
1	2	1	91
5	2	2	106
10	2	3	109
20	3	1	103
35	4	1	121
50	5	1	140
70	5	2	145
100	10	3	150
125	12	1	310
150	10	1	240
175	15	2	587
200	17	3	656
225	20	2	688
250	20	1	778

Table 1. Processing Time

From the data in table 1, we conclude that the time for pre-processing is directly proportional to the number of series and weight of window for which streaming data is present. The processing time also depends on the number of time intervals for which data is present and number of analysis variables; however number of hierarchy levels does not impact this time.

The performance has been tested with 360000 records with data for 200+ data streams.

3.Data Point Plotting Time

No of Series	No. of Data Points	Time for plotting in milliseconds per data point		
		Best	**Average**	**Worst**
1	2000	4.32	5.29	6.27
3	6000	2.17	2.54	2.92
5	10000	49.09	68.48	87.87
10	20000	62.05	74.02	85.99

Table 2. Plotting Time

Conclusion:

- Time for plotting depends on the number of the data points to be plotted.
-

4.Summarization Accuracy

Weighted Moving Average and Exponential smoothing with trend are used for calculating summarization. MAPE (Mean Absolute Percent Error) is used to detect the accuracy. Less the MAPE % more accurate is summarization.

Figure 2 shows comparative analysis of both the methods.

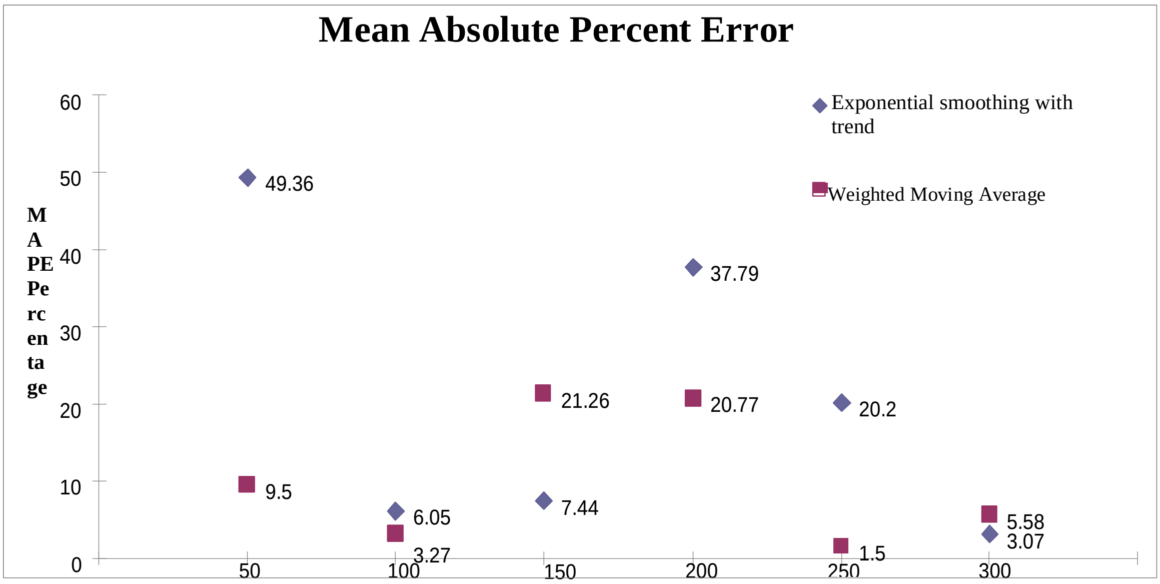


Fig. 2. Summarization error chart

From the chart above, it is observed that Weighted Moving Average has MAPE <= 20% and thus provides >= 80% accuracy.

Time series visualization system uses Weighted Moving Average will be able to provide 80% to 98.5% accurate summarization. If summarization is done in forward horizon, system could forecast the arriving data points with accuracy >=75%. Authors mention that 75% accurate results are acceptable [21].

5. User Interface developed for Visualization:

The following charting components are developed to visualize an Amnesic data being monitored with monitoring view (fig. 3 & 4).

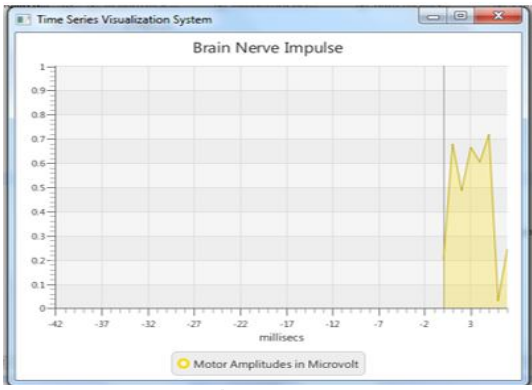


Fig.3.Stream Monitoring Chart View 1

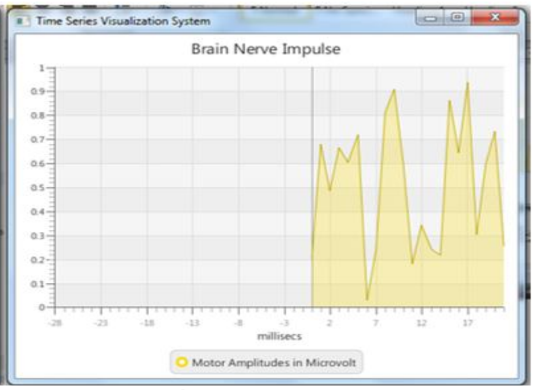


Fig. 4. Stream Monitoring Chart View 2

Different streams are analysed using comparative view (fig 5 & 6).

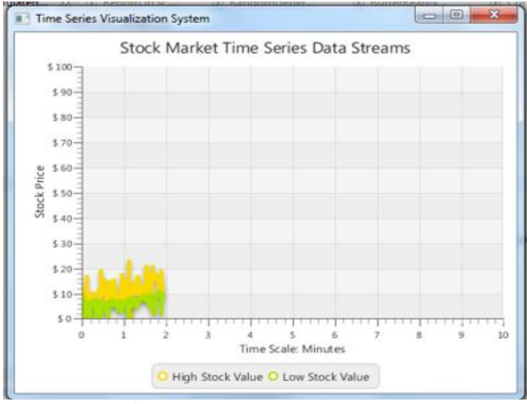


Fig.5.Amnesic Chart View 1

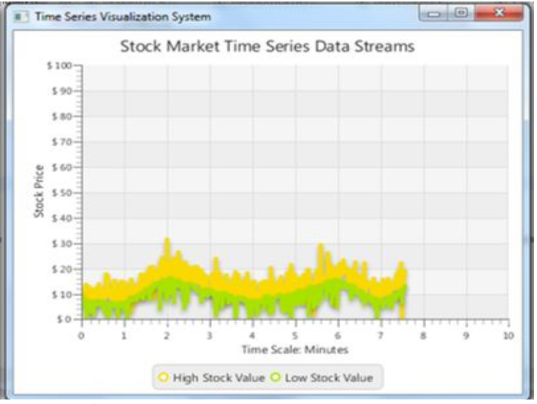
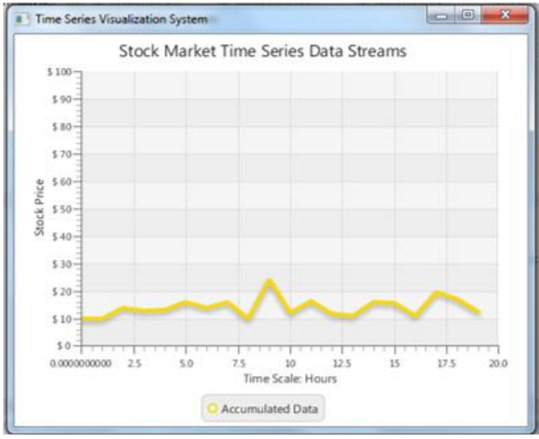


Fig.6. Amnesic Chart View 2

Accumulated series are displayed using summarization view (fig.7 & 8).



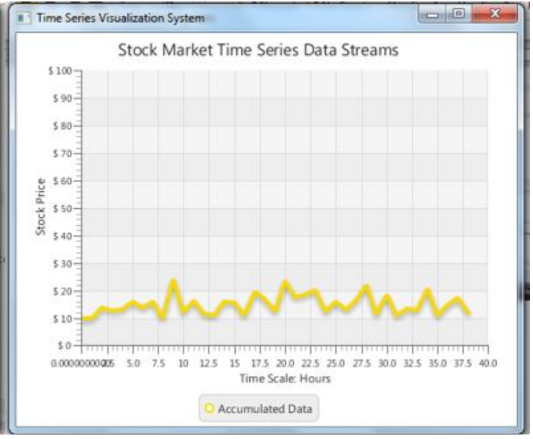
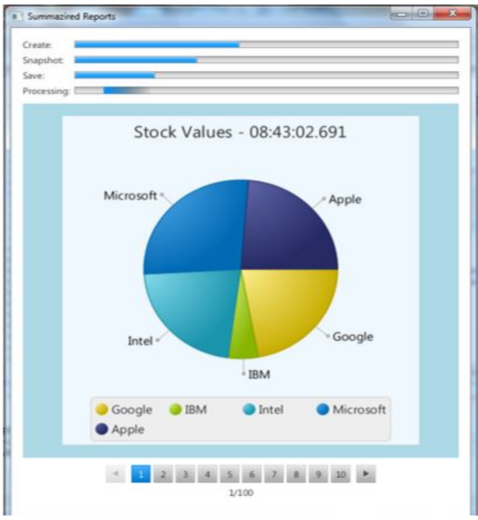


Fig.7.Accumulated Chart View 1 Fig. 8. Accumulated Chart View 2

For analysis of various streams, report view is developed (fig 9 & 10).



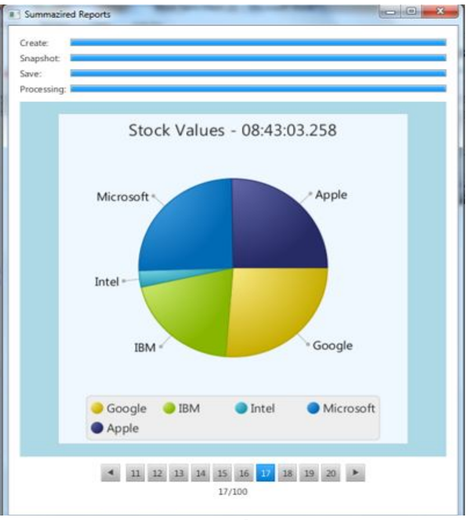


Fig.9Analysis Report View 1

Fig. 10. Analysis Report View 2

V. Conclusion and Future Work

The runtime view updating capability is lacking in many systems as these systems provide the static view in terms of charts; in contrast our system is dynamic and helps users in visual exploration. Accumulation functionality is new and not available in any other visualization systems.

Many tools exist for visual exploration; however our system provides capability to summarize in financial context which would be extremely useful for business analysts in decision making. Time series visualization system provides minimal display time by data pre-computing and hierarchical storage.

Summarization of data streams and accumulation of data points is implemented using weighted moving average with trend and such methodology is prerequisite for forecasting techniques. This is the default algorithm used by the system and user cannot select the algorithm to be used. System can be enhanced to provide the facility to choose forecasting algorithm and compare the results.

Future approach of the system would be to stream the visual on various portable devices such as mobile phones, tablets, desktops and personal computers.

Acknowledgment

First Author express gratitude to his project manager Dinesh Apte for the useful comments, remarks and engagement through the learning process of this part of master thesis. This research is sponsored by SAS Research & Development (India) Pvt. Ltd.

References

[1] Aggarwal, Charu C., ed. Data streams: models and algorithms. Vol. 31. Springer, (2007).
[2] Arabie, Phipps, and Lawrence J. Hubert. "An Overview of Combinatorial Data Clustering and Classification". (1996).
[3] Barbará, Daniel. "Requirements for clustering data streams." ACM SIGKDD Explorations Newsletter 3, Vol. 2, pp. 23-27, (2002).
[4] Babcock, Brain, MayurDatar, Rajeev Motwani, and Liadan O'Callaghan. "Maintaining variance and k-medians over data stream windows." In Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, pp. 234-243. ACM, (2003).
[5] Chen, Yixin, and Li Tu. "Density-based clustering for real-time stream data." In Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 133-142. ACM, (2007).
[6] Gama, Joäao, and Mohamed MedhatGaber, eds. Learning from data streams: processing techniques in sensor networks. Springer, (2007).
[7] Gama, Joao, Pedro Pereira Rodrigues, Eduardo J. Spinosa, and André Carlos Ponce Leon Ferreira de Carvalho. Knowledge discovery from

data streams. London: Chapman & Hall/CRC, (2010).

[8] Fayyad, Usama, Gregory Piatetsky-Shapiro, and Padhraic Smyth. "From data mining to knowledge discovery in databases." AI magazine 17, no. 3, (1996).

[9] Guha, Sudipto, Adam Meyerson, Nina Mishra, Rajeev Motwani, and Liadan O'Callaghan. "Clustering data streams: Theory and practice." Knowledge and Data Engineering, IEEE Transactions on 15, no. 3, pp. 515-528, (2003).

[10]Golab, Lukasz, and M. Tamer Özsu. "Issues in data stream management."ACMSigmod Record 32, no. 2, pp. 5-14, (2003).

[11] Ren, Jiadong, and Ruiqing Ma. "Density-based data streams clustering over sliding windows." In Fuzzy Systems and Knowledge Discovery, 2009. FSKD'09. Sixth International Conference on, Vol. 5, pp. 248-252. IEEE, (2009).

[12] Zhou, Aoying, Feng Cao, Weining Qian, and CheqingJin. "Tracking clusters in evolving data streams over sliding windows." Knowledge and Information Systems 15, no. 2, pp. 181-214, (2008).

[13] Zhu, Yunyue, and Dennis Shasha. "Statstream: Statistical monitoring of thousands of data streams in real time." In Proceedings of the 28th international conference on Very Large Data Bases, pp. 358-369. VLDB Endowment, (2002).

[14] Keogh, Eamonn, Selina Chu, David Hart, and Michael Pazzani. "An online algorithm for segmenting time series." In Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on, pp. 289-296. IEEE, (2001).

[15] Babcock, Brian, MayurDatar, and Rajeev Motwani. "Sampling from a moving window over streaming data." In Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms, pp. 633-634. Society for Industrial and Applied Mathematics, (2002).

[16] Golab, Lukasz, David DeHaan, Erik D. Demaine, Alejandro Lopez-Ortiz, and J. Ian Munro. "Identifying frequent items in sliding windows over on-line packet streams." In Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement, pp. 173-178. ACM, (2003).

[17] Boehm, B.W., “Software risk management: principles and practices”, Software, IEEE, Jan 1991, Vol. 8, Issue 1, pp. 32-41, (1991).

[18] Roger S. Pressman, Software Engineering – A practitioners Approach, 6th Edition (1992).

[19] Datar, Mayur, Aristides Gionis, Piotr Indyk, and Rajeev Motwani. "Maintaining stream statistics over sliding windows." SIAM Journal on Computing 31, no. 6, pp. 1794-1813, (2002).

[20] Gibbons, Phillip B., and SrikantaTirthapura. "Distributed streams algorithms for sliding windows." In Proceedings of the fourteenth annual ACM symposium on Parallel algorithms and architectures, pp. 63-72. ACM, (2002).

[21] Wilschut, Annita N., and Peter MG Apers. "Dataflow query execution in a parallel main-memory environment." Distributed and Parallel Databases 1, no. 1, pp. 103-128, (1993).

[22] Kang, Jaewoo, Jeffrey F. Naughton, and Stratis D. Viglas. "Evaluating window joins over unbounded streams." In Data Engineering, 2003. Proceedings. 19th International Conference on, pp. 341-352. IEEE, (2003).

[23] Golab, Lukasz, and M. Tamer Özsu. "Processing sliding window multi-joins in continuous queries over data streams." Proceedings of the 29th international conference on Very large data bases, Vol. 29. VLDB Endowment, (2003)

[24] Koski, Antti, MarttiJuhola, and MerikMeriste. "Syntactic recognition of ECG signals by attributed finite automata." Pattern Recognition 28.12, (1995).

[25] Vullings, H. J. L. M., M. H. G. Verhaegen, and Henk B. Verbruggen. "ECG segmentation using time-warping." In Advances in Intelligent Data Analysis Reasoning about Data, pp. 275-285. Springer Berlin Heidelberg, (1997).

[26] Qu, Y., Wang, C. & Wang, S. “Supporting fast search in time series for movement patterns in multiples scales.” Proceedings of the 7th International Conference on Information and Knowledge Management, (1998).

[27] Wang, Changzhou, and X. Sean Wang. "Supporting content-based searches on time series via approximation." In Scientific and Statistical Database Management, 2000. Proceedings. 12th International Conference on, pp. 69-81. IEEE, (2000).

[28] Shatkay, Hagit, and Stanley B. Zdonik. "Approximate queries and representations for large data sequences." In Data Engineering, 1996. Proceedings of the Twelfth International Conference on, pp. 536-545. IEEE, (1996).

[29] Park, Sanghyun, Dongwon Lee, and Wesley W. Chu. "Fast retrieval of similar subsequences in long sequence databases." In Knowledge and Data Engineering Exchange, 1999. (KDEX'99) Proceedings. 1999 Workshop on, pp. 60-67. IEEE, (1999).

[30] Keogh, Eamonn, Kaushik Chakrabarti, Michael Pazzani, and Sharad Mehrotra. "Dimensionality reduction for fast similarity search in large time series databases." Knowledge and information Systems 3, no. 3, pp. 263-286, (2001).

[31] Palpanas, Themis, Michail Vlachos, Eamonn Keogh, and DimitriosGunopulos. "Streaming time series summarization using user-defined amnesic functions."Knowledge and Data Engineering, IEEE Transactions on 20, no. 7, pp. 992-1006, (2008).

[32] Silva, Jonathan A., Elaine R. Faria, Rodrigo C. Barros, Eduardo R. Hruschka, André CPLF de Carvalho, and João Gama. "Data stream clustering: A survey." ACM Computing Surveys (CSUR) 46, no. 1 (2013).

[33] Aigner, Wolfgang, Silvia Miksch, Wolfgang Muller, Heidrun Schumann, and Christian Tominski. "Visual methods for analyzing time-oriented data." Visualization and Computer Graphics, IEEE Transactions on 14, no. 1, pp. 47-60, (2008).

[34] Kaushal Chauhan, Mukta Takalikar, Dinesh Apte. “Visualization of Time Series Data Streams.” In International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE), Vol. 3, pp. 879-891, December (2013).

BIOGRAPHY

Kaushal Lalitkumar Chauhanis a PG Student in the Computer Engineering Department, College of Pune Institute of Computer Technology, University of Pune, India and as a Research Intern in SAS Research & Development, (India) Pvt. Ltd. He received Bachelor of Computer Science & Engineering (B.E.) degree in 2011 from BAMU, Aurangabad, MS, India. His research interests are Computer Graphics, Visualization, Big Data, Internet of Things, etc.