Continuous Prediction of Closed Frequent Itemsets from High Speed Distributed Data Streams using Parallel Mining on Manifold Windows with Varying Size

V.Sidda Reddy
Professor, CSE
Sagar Institute of Technology,
Hyderabad, India.

T.V. Rao
Professor, CSE
PVP Siddhartha Institute Tech,
Vijayawada, India.

A.Govardhan
Professor, SIT
JNTUH
Hyderabad, India

ABSTRACT
Continuous prediction of closed frequent itemsets from high speed distributed data streams is an active research work, which is because of the conflict to the process time taken to perform mining consistent itemsets from current records and high alacrity transmission time in data streams. By the motivation gained from our earlier proposed models, here we devised a novel closed frequent itemset mining model for high speed distributed data streams. The said model is referred as Parallel Closed Frequent Itemsets Mining (PCFIM) over High Speed Distributed Data streams by Manifold Varying Size Windows (MVSW). The results obtained from experiments are significant to prove that the proposed PCFIM is scalable and robust on high speed data streams and miles ahead over existing bench mark models.

Keywords
Data Streams, Distributed Data Stream, Closed Frequent Itemsets Mining, Sliding Window, Varying Window.

1. INTRODUCTION
More recently, the need to process continuous and high speed data has motivated the field of data stream mining. The data streams are continuous, unbounded usually come with high speed in sequence. The data stream signifies the input data which happens with a sudden rate that is most likely absolute. Traditional data mining techniques can not be easily applied to data streams mining due to unique characteristics of data streams [23] such as continuous, high speed, unbounded, time-varying and sequence.

Instances for channels to data streams consist of prospect push streams, system tracking information, telephone recording calls, sensor network, scientific data, and web pages and so on. Owning to huge number of in- succeeding information on data stream mining, information aspects should be read simply after having the constrained number of foremost memory inside the mining procedure [13]. Concerning each performs of information mining, consistent itemset mining [1] in excess of data streams appeals to a lot of desire into data mining group because it could show effective info to different programs. Moving pane is an absorbing replica for consistent itemset mining more data streams [6] [7] [8] [9] [14] [15] [16] [17] [19]. During this method, exclusively most in recent times appeared transactions tend to be viewed for that mining use. Sliding window method deals with notion substance inside and contribute information stream with getting simply current operations. Subsequently, that window usually consist latest approach. In consistent pattern mining crisis, the idea consults to a collection of consistent patterns. Generally you can find two various techniques in data flow mining, in managing model modify in the window. 1st, finding approach enhances as well as retrieving from it and the 2nd, changing that mining consequences towards newer approach when time proceeds. The 2nd strategy is much effective also commonly worn to sequence mining across data streams considering it won't necessitate a self-governing method of alter recognition. An additional asset to sliding window is the restricted range in memory consumption as well as handling energy. In an order aware window, every sliding can managed with placing a fresh transaction also removing the earliest transaction. In that approach, executing single sliding process not substantially impact a window composition and mining consequences however requires the significant moment to enhance each of those. With enhancing an appliance of supplement and removal with a group of connections as better to splitting overlapping windows through separate panels, the sliding window procedure could carried out by window acquisition and removal. The splitting window improves the endeavor to sliding window that much longer when there are adequate connections for every window. That is because of utilizing a similar procedure in the window information and mining consequences for a few connections rather than single. Additionally, we will make a data construction to conveniently accumulate panes of the window.

Within a time-varying data stream, the prevalence of itemsets might alter with time. This might remove many old frequent itemsets and also propose new ones. When adjustments are recognized (e.g., a spacious fraction of old frequent itemsets are noticed no extended valid over the test opposing new approaching data), the consistent itemsets desire to be re-computed to replicate the specifications of new data as well as the result must be obtainable as soon as possible.

In this paper, we investigate the problem of finding closed frequent itemsets in distributed high speed data streams from a different angle, which concentrates to minimize the required disk space and processing time and maximize the accuracy of mining closed frequent itemsets. It is well-known that finding accurate closed frequent itemsets needs multiple scans on the data. Our basic idea is distributed two levels approach that in initial level scans a window to track the locally frequent itemsets and then in the second level, updates the global set of frequent itemsets and their support. Especially, we use our earlier approach TIFIM [21] in first level of the devised model, which is a tree based incremental frequent itemset mining model for data streams. The BIDE [5] rule is used in second level to identify the globally closed frequent itemsets. A max-K clusters formation by Jaccard similarity [20] is personalized to define variable size and time windows. This

clustering process is derived from our earlier proposed model called MFI-VWS-CVA [22]. As the incremental approach for mining frequent itemsets from a given window is being used, the computation cost is dramatically reduced and due to the clustering process used to form the variable size windows, the reusability of the disk space will be improved, hence the usage of minimal disk space by the devised model is our another significant claim.

The rest of this paper is organized as follows. Section 2 presented the associated research in mining frequent itemsets over data streams. The proposed approach is explored in the Section 3, which followed by section 4 that delivers the preliminary experiment results. Finally, the summary of the article is discussed in section 5.

2. RELATED WORK

The crisis of consistent itemset mining during the data supply would be unveiled with Manku et al. [2] precisely the writers projected algorithms to consistency counts in accessories through data streams. Later, they offer their efforts to a frequent itemsets group and test strategy during data streams. Through the endure decade, many specific as estimated algorithms already projected for mining consistent patterns concerning data avenues. Depending on data flow control replica, that they are specified into three distinctive classifications, milestone built [3] [10] [11] damped otherwise time crumble situated [12] [18] also sliding window modeled [6] [7] [8] [9] [14] [15] [16] [17] [19] algorithms. Within the milestone unit, arriving data from the position period known as landmark till the existing time will be considered. That landmark might be building or resuming time of this strategy. At this method, preceding as well as current transactions of effort stream tend to be known as becoming similar. DSM-FI [3] is the landmark built algorithm. In that particular algorithm each transaction is transformed to modest operations also placed to a synopsis data structure known as item-suffix consistent itemset plan, and is built upon a prefix-tree. At [10] the writers worn a Chernoff-Bound for build the estimated outcome of frequent sequences more in the landmark window. Zhi-Jun et al. [11] taken a lattice design, mention to a consistent specify tree, which split into multiple equal sessions of accumulated patterns using similar transaction-ids in the particular group. Frequent patterns tend to be split up to identical classes, also such frequent patterns which symbolize both edges of every class are controlled, an another frequent patterns tend to be trimmed. As part of the time decline replica, the significance in the data aspects diverges built on the release an effort to stress newly emerged data. Chang as well as Lee recommended an algorithm described as estDec considering time decay model where every transaction includes mass reducing through age [12]. In that approach to diminish an outcome to old transaction in specify of consistent patterns a change rank is described. At [18] an algorithm identical with the estDec is endorsed to mining optimum consistent itemsets quickly in data stream built upon damped method. Generally there is a bit in sliding window located algorithms suggested to frequent itemset mining on data streams. Because the projected algorithm performs the sliding window model as well as this mold is a commonly applied model to data stream mining, below massive algorithms projected in that replica are analyzed. DSTree [9] as well as CPS-Tree [15] will be two algorithms to choose a prefix tree towards accumulate rare transactions in the sliding window. DSTree takes an unchanging tree construction in canonical set up to branches such as CPS-Tree a prefix tree structure can regained of manage picking memory consumption. Both of [9] as well as [15] execute the mining process applying FP-Growth [4] algorithm which had been projected for inert databases. At [16] an algorithm mainly MFI-TransSW turned out to be recommended that is depending upon the Apriori algorithm [1]. Such algorithm mines each consistent itemsets on current window of connections. This takes the little sequence for each item to store their prevalence information inside the window. At [19] the sliding window centered algorithm has been recommended where window desire is vigorously kept having some basic catalogs.

Lin et al. [6] projected another technique for mining consistent patterns more on time sensitive sliding window. Within specific approach, that window is split into many batches that kind of itemset mining was carried out independently. One Instance algorithm [8] confirms sealed frequent itemsets by keeping the boundary around frequent enclosed itemset also some another itemsets. The SWIM [14] is a window built algorithm that kind of consistent itemsets at one pane of the window tend to be viewed as in addition evaluation to obtain frequent itemsets to entire in the window. This provides a uniting of frequent patterns to many panes also gradually updates any holds and trims periodic data. That keeps transactions on the window type for the prefix tree of every panel. At [17] the writers constructed an algorithm on mining non-derivable consistent itemsets in data streams. That algorithm frequently keeps non-derivable consistent itemsets of a moving window. Non-derivable as well as restricted frequent itemsets tend to be certain kinds of frequent itemsets that are noticed as a summary of most consistent itemsets.

Chang as well as Lee projected the estWin algorithm [7] which seeks current consistent patterns adaptively beyond transactional information streams having a sliding window replica. Because the algorithm is accurately associated with the estWin algorithm, it can be illustrated much more specifics. That algorithm utilizes a tracking lattice for the kind of the prefix tree towards control the number of frequent itemsets across the data stream. Every node in the lattice signifies an itemset which are formulated with items kept in nodes on the direction in the root to a node. Past node of this route keeps on content regarding the itemset, e.g., assistance. This algorithm utilizes a decrease minimal assist called minimum important to initial recognize more frequent itemset also to adept determine their assistance. Whenever a new transaction can appeared in the input data stream then recent collection of frequent itemset can modified with viewing relevant itemset in the tracking lattice. Perhaps unique considerable itemsets tend to be recognized with second traversal in the associated routes of the tracking lattice utilizing the operation. That algorithm keeps a change of transactions in the window with the storage to overcome its consequence whenever these grown to be terminated. For eliminating earliest transaction during that window, associated routes of the tracking lattice should be called anymore. That algorithm, snips measly itemset in the tree after moving the frequent range of transactions utilizing an autonomous procedure identified the compel pruning. The itemset is placed towards a prefix tree after many of its subsets grown to be appreciable and are accumulated into the tree

That algorithm shows an assistance of another appreciable itemset inside the preceding transactions in the window. Concerning an itemset possessing length k (k-itemset), the calculated assist is adequate to smallest assist amongst its subsets with span k-1. Every node of this prefix tree comprises insight using probable count (pcnt), actual count

(acnt), error (err) also foremost transaction (mid). The pcnt points to the calculable occurrence of the itemset concerning each preceding operations of the window recently an itemset can placed towards prefix tree. The acnt may supervise occurrence of the itemset after it is placed with the tree. The error will be mistake of the determined assistance of this itemset estimated utilizing their subsets. The mid is an id of operation that triggers this itemset is appended towards prefix tree. Generally assistance on the itemset for the prefix tree can calculated with the summing up of its acnt as well as pcnt. For further ideas about processing the potential count also the error within the assistance to considerable itemsets as well as other factors for the estWin algorithm, the planning viewers can direct on [7]. After supplement of an itemset, that new orders appears as well as old transactions tend to be eliminated in the window and oversight for the assistance of this itemset would be condensed. The removal of error is caused by eliminating the result of old transactions also removal in the appeal of potential relies.

The sliding window formulated consistent itemsets mining on data streams might get classified into two major groups. During the foremost group [9] [15] [16] [19] a window can held through existing transactions also the mining procedure could extort consistent routines inside the active window whenever the customer proposes a request. So, the objective is to accumulate operations with lower memory intake and carrying out the mining strategy effectively. During the 2nd group [6] [7] [8] [14] [17] the mining outcomes are continually upgraded by placing new operations towards window and extracting previous operations from window. Hence, quick modernizing for the mining consequences as well as lower memory utilization to the mining consequence also operations of this window tend to be significant. Algorithms of this 2nd group become much beneficial for consumer as they may observe that mining outcome promptly when it's requested. In rapid process, an estimated algorithm that determine group of consistent itemsets inside sliding window using extreme standard of this outcome may appropriate. Algorithms on the 1st group [9] [15] [16] [19] will not adaptively preserve also improve that mining outcome. Thus, following the mining, whenever unique operations tend to be appeared with the supply, acquired outcome gets invalid for the consumer thereby the mining process should be re-executed. However, employing a mining algorithm all in all of the window requires significant processing and also memory specifications particularly in significant window size pertaining to algorithms of 2nd group [6] [8] [14] [17] [19] that the mining outcomes are modified adaptively. Anyhow, the efficiency of such algorithms is comparatively low in contrast to maximum arrival pace of feedback channels. As a result, on this analysis, an effective closed frequent itemset mining through high speed dispensed data streams by manifold adjustable size also time windows is projected for adaptive upkeep of closed consistent itemsets

3. PARALLEL CLOSED FREQUENT ITEMSETS MINING (PCFIM) OVER HIGH SPEED DISTRIBUTED DATA STREAMS BY MANIFOLD VARYING SIZE WINDOWS (MVSW)

The data: Let 'S' be the set streams defined as $S=\{s_1,s_2,s_3,....s_i,s_{i+1},.....s_n\}$, such that each stream $\{s_i \forall s_i \in S\}$ streams data with varying speed. Let tr be the transaction record transmitted by a stream $\{s_i \forall s_i \in S\}$ as one unit.

The Mining Model: Let Mining Model MM on the other side connected to set of streams S that defined as shown in following equation, and accepting transaction records transmitted by all these streams.

$$\bigcup_{i=1}^{|S|}\{s_i \forall s_i \in S\}$$

The approach: Upon receiving a transaction tr by the mining model MM, checks using jaccard similarity that if any existing window suits to adapt the transaction tr. If so then moves that transaction to the appropriate window, otherwise forms a new window with tr as centroid. If the number of windows is found to be greater than k, then initiates the process of merging two or more windows into a single cluster by measuring their context similarity [22]. During this a new window that formed due to merging of two or more windows will retain the most recent update time of the window involved in merging process as its recent update time. Further it initiates finding locally frequent itemsets from windows that are having difference between current time and recent update time below the given threshold. In regard to find the locally frequent itemsets from selected windows in distribute manner, the mining model uses the tree based incremental mining for frequent itemsets TIFIM [21]. Further the global closed frequent itemsets will be updated by adapting local frequent itemsets under bide rule [5].

3.1 Algorithmic approach of the PCFIM

Begin

Let "S" be the set of streams that can be defined as $S=\{s_1,s_2,s_3,....s_i,s_{i+1},.....s_n\}$

Let tr be a transaction received at a point of time

Let 'W' be the set of windows currently in scope, which can be defined as $W=\{w_1,w_2,w_3,....w_i,w_{i+1},.....w_m \forall m \le k\}$. Here in this equation 'k' is maximum number of windows threshold.

$MM \leftarrow \{s_i : tr \forall s_i \in S\}$ This equation indicates that the mining model MM is receiving a transaction from stream s_i

Upon receiving tr by MM it performs the following

Let $dst \leftarrow \infty$ and $wid \leftarrow 0$

For each window $\left\{ w_i \forall \bigcup_{i=1}^{|W|}(w_i \in W) \right\}$ begin

$d = \text{findDistance}(tr, w_i)$

If $d < \delta$ then begin // Here in this equation δ minimal distance threshold

If $d < dst$ then begin

$dst \leftarrow d$

$wid \leftarrow i$

End

End

End

If $dst \neq \infty \&\& wid \neq 0$ then Begin

$w_{wid} \leftarrow tr$ // the distance between w_{wid} and tr is minimal and less than the given distance threshold, hence moving transaction tr to window w_{wid}

End

Else Begin

If $|W| < k$ then begin

$W \leftarrow \{w_{|W|+1} \forall [w_{|W|+1} \leftarrow tr]\}$ //New window $w_{|W|+1}$ with transaction tr as centroid is formed

End

If $|W| \geq k$ then begin

$mergeWindows(W)$

For-each window $\left\{ w_i \forall \bigcup_{i=1}^{|W|} w_i \in W \right\}$ Begin

If $(crt - rut(w_i)) > \tau$ then begin //Verifying the relation between threshold τ and difference among current time crt and recent update time $rut(w_i)$

$FIS(w_i) = TIFIM(w_i)$ // Finding locally frequent itemsets from window w_i using TIFIM [21] technique

$updateGCFIS(FIS(w_i))$ // updateGCFIS is a method updates globally closed frequent itemsets by adding local frequent itemsets $FIS(w_i)$ under BIDE rule [5].

End; End

End

End

End

3.2 Algorithmic approach of findDistance(tr, w_i)

Begin

$attrSet = \phi$ // $attrSet$ is attribute set with ' φ 'as initial value

For-each transaction $\{tr(w_i) \forall tr(w_i) \in w_i\}$ begin

$attrSet \leftarrow attrSet \bigcup tr(w_i)$ //moving all attributes of transaction $tr(w_i)$ to $attrSet$ that are not in $attrSet$

End

$d = 1 - \frac{|attrSet \bigcap tr|}{|tr|}$ //Finding the distance between attribute set of window w_i and transaction tr

Return d

End

3.2 Algorithmic approach of TIFIM [21]

- A bush represents itemsets with two attribute pair such that these attributes belongs to fs_{tran} and transactions contain that pair.
- The coverage to measure the frequency of the itemsets can be considered and set in the context of window w_{tran} size.
- The coverage of two attribute itemsets can be the count of number of Childs in a bush represented by each pair of attributes.
- An asynchronous parallel process called frequent itemsets finder (FIF) performs as follow:
- Initially picks the bushes with coverage more than given coverage threshold **COV** .
- Prepare new bushes from each two bushes by union the roots and intersects the Childs, and retains it if new bush coverage is greater or equal to **COV** else discards.
- This continues until no new bush formed.

The pruning process: A bush b_i said to be sub-bush to bush b_j if $r_{b_i} \subseteq r_{b_j}$ and $\mathbf{COV}_{(b_i)} \leq \mathbf{COV}_{(b_j)}$. Since sub-bush b_i represented by b_j , then bush b_i can be pruned from the bush-set B .

Find frequent items: At an event of time, frequent itemsets can be found as follows

The roots of the bushes with coverage more than given **COV** can be claimed as frequent itemsets.

A bush ' b_i ' coverage can be find as follows

If a bush b_j found to be such that $b_i \subseteq b_j$ and coverage value of b_j is higher than any other bush b_k such that

$b_i \subseteq b_k$, then the coverage of b_i said to be $\text{COV}_{(b_j)} + \text{COV}_{(b_i)}$.

Caching processes : Input - At an event of time a window w_i with transaction t_i received

For each transaction t_i :

$$\{(a_1, a_2, a_3, ...a_i)\forall$$

Let set of attributes $(a_1, a_2, a_3, ...a_i) \in t_i \wedge$

$$(a_1, a_2, a_3, ...a_i) \subseteq A_{set}\}$$

For each pair of attributes $\{(a_m, a_n)\forall(a_m, a_n) \in t_i\}$, if found a bush $\{b_i \exists (a_m, a_n)\ as\ root\}$ then add transaction t_i as node to bush b_i , else prepare a bush such that $\{b_i \exists (a_m, a_n)\ as\ roo\,t \wedge t_i\ as\ node\}$

FIF: The bush set B prepared by caching process is said to be input to FIF

For each bush $\{b_i \forall b_i \in B\}$ perform the following:

For each bush $b_{i+c} \exists (c := 1, 2, 3.....n) \wedge b_{i+c} \in B)$

Forms a bush $\{b_{(i \cup i+c)} \exists b_{(i \cup i+c)} \notin B\}$ by Union the roots of the ' b_i 'and ' b_{i+c} ' ($r_{(b_i)} \bigcup r_{(b_{i+c})}$) and intersects nodes of b_i and b_{i+c} ($ts_{(b_i)} \bigcap ts_{(b_{i+c})}$).

3.3 Algorithmic approach of *updateGCFIS*($FS(w_i)$)

Begin

For-each frequent itemset $\{fis \forall fis \in FS(w_i)\}$ Begin

If $fis \in GCFIS$ then // *GCFIS* is globally closed frequent itemsets

Begin

$GCFIS(fis)_{\text{sup}} = GCFIS(fis)_{\text{sup}} + fis_{\text{sup}}$ // adding the support of the frequent itemset of window w_i

End

For-each closed frequent itemset $\{cfis \forall cfis \in GCFIS\}$ // *GCFIS* is globally closed frequent itemsets

Begin

For-each closed frequent itemset $\{ocfis \forall ocfis \in GCFIS\}$ Begin //applying BIDE rule [5]

If ($cfis \cong ocfis$) then *continue the loop*

Else if $cfis \subset ocfis \,\&\&\, cfis_{\text{sup}} \cong ocfis_{\text{sup}}$ then delete $cfis$ from $GCFIS$

End

End

End

End

4. EMPIRICAL STUDY OF THE PCFIM

The experiments were conducted on a set of synthetic datasets generated with following constraints:

- The max length of the record varies from 5 to 45.
- The min length of the record ranges from 3 to 7.
- The max frequency of each attribute ranges from 1% to 35%.
- The size of the dataset ranges from 10000 to 50000 that streamed from distributed sources.

The sparseness of the dataset is achieved with a low max length of the record and low frequency of the each attribute, but size is not a constraint, in contrast the dense dataset is formed with constraints (i) high max and min length of the record, (ii) and high frequency of the attributes.

The Experiments conducted under the following constraints:

- Distribution of the streams ranges from 3 to 5.
- The max window count k ranges from 2 to 10.
- Similarity threshold $\eta(tr \leftrightarrow w)$ ranges from 0.7 to 0.925.
- Elapsed recent window update time threshold τ ranges from 0.25 sec to 1second.
- The streaming speed ' σ ' is set in range of 0.01 seconds to 0.05 seconds.

The experiment results indicating that k and σ are highly correlated and PCFIM delivers scalable performance during high speed streaming of dense data set is under maximal k and maximal $\eta(tr \leftrightarrow w)$ values. The streaming speed σ and denseness ϑ of the dataset are not correlated to influence the performance of the PCFISM. The Fig 1 and Fig 2 indicating the correlation of k and $\eta(tr \leftrightarrow w)$ in completion time scalability and memory usage respectively, which is during high speed distributed streaming of dense data. As the result explored in Fig 1, the maximum k and maximum $\eta(tr \leftrightarrow w)$ leads to scalable completion time, and in similar way the maximal k and $\eta(tr \leftrightarrow w)$ leads to use minimal disk slot that explored in result figured by Fig 2.

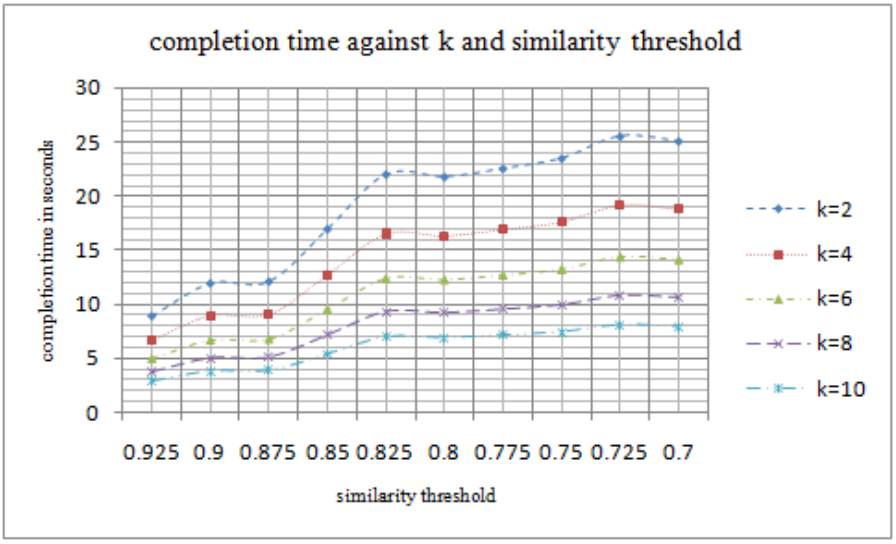

Fig 1: Time taken to complete the process against divergent k and similarity threshold values

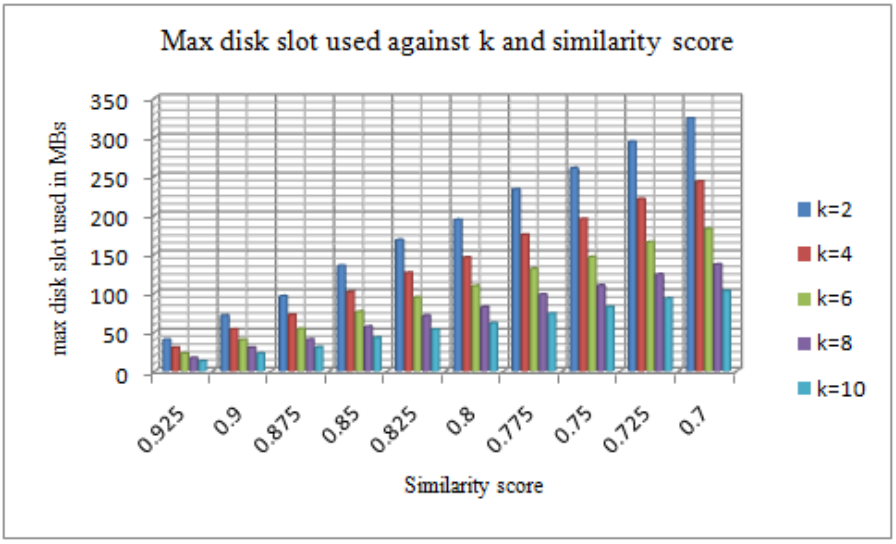

Fig 2: max disk slot used against divergent k and similarity threshold values

5. CONCLUSION

This paper explored a parallel sealed frequent itemset mining during high speed distributed data flows, which is from the motivation of our earlier works TIFIM [21] and MFI-VWS-CVA [22]. The experimental results indicating that manifold varying size maximal k window formation is an effective strategy to achieve scalable performance during continues detection of closed frequent itemsets from high speed distributed data streams. This window formation is an approach devised by personalizing the k-means clustering strategy. In the future experiments need to be conducted to compare the performance of the PCFISM with other bench mark strategies. Also it will be interesting to extend this work, when record streaming is done by partitioning vertically between multiple streams.

6. ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their valuable comments. We also thank the authors of all references for helping us setup the paper.

7. REFERENCES

[1] R.Agrawal and R. Srikant, "Fast algorithms for mining association rules", In Proceedings of International Conference on Very Large Databases, 1994, pp. 487-499.

[2] G. S. Manku and R. Motwani, "Approximatefrequency counts over data streams", In Proceedings of VLDB International Conference on Very Large Databases, 2002, pp. 346-357.

[3] H.-F. Li, S.-Y. Lee, and M.-K. Shan, "An efficient algorithm for mining frequent itemsets over the entire history of data streams" , In Proceedings of International Workshop on Knowledge Discovery in Data Streams, 2004, pp. 20-24.

[4] J. Han, J. Pei, Y. Yin, and R. Mao, "Mining frequent patterns without candidate generation: A frequent-pattern tree approach", Data Mining and Knowledge Discovery, Vol. 8, 2004, pp. 53-87.

[5] J. Han, J. Pei, Y. Yin, and R. Mao, "Mining frequent patterns without candidate generation: A frequent-pattern tree approach", Data Mining and Knowledge Discovery, Vol. 8, 2004, pp. 53-87.

[6] C. H. Lin, D. Y. Chiu, Y. H. Wu, and A.L. P. Chen, "Mining frequent itemsets from data streams with a time-sensitive sliding window" , In Proceedings of SDM International Conference on Data Mining, 2005, pp. 68-79.

[7] J. H. Chang and W. S. Lee, "estWin: Online data stream mining of recent frequent itemsets by sliding window method", Journal of Information Science, Vol. 31, 2005, pp. 76-90.

[8] Y. Chi, H. Wang, P. S. Yu, and R. R. Muntz, "Catch the moment: maintaining closed frequent itemsets over a data stream sliding window", Knowledge and Information Systems, Vol. 10, 2006, pp. 265-294.

[9] C. K. S. Leung and Q. I. Khan, "DSTree: a tree structure for the mining of frequent sets from data streams", In Proceedings of IEEE International Conference on Data Mining, 2006, pp. 928-932.

[10] J. X. Yu, Z. Chong, H. Lu, Z. Zhang, and A. Zhou, "A false negative approach to mining frequent itemsets from high speed transactional data streams", Information Sciences, Vol. 176, 2006, pp. 1986-2015.

[11] X. Zhi-Jun, C. Hong, and C. Li, "An efficient algorithm for frequent itemset mining on data streams" ,In Proceedings of the 6th Industrial Conference on Data Mining, 2006, pp. 474-491.

[12] J. Chang and W. Lee, "Finding recently frequent itemsets adaptively over online transactional data streams", Information Systems, Vol. 31, 2006, pp. 849-869.

[13] J. Han, H. Cheng, D. Xin, and X. Yan, "Frequent pattern mining: current status and future directions", Data Mining and Knowledge Discovery, Vol. 15, 2007, pp. 55-86.

[14] B. Mozafari, H. Thakkar, and C. Zaniolo, "Verifying and mining frequent patterns from large windows over data streams", In Proceedings of International Conference on Data Engineering, 2008, pp. 179-188.

[15] S. K. Tanbeer, C. F. Ahmed, B. S. Jeong, and Y. K. Lee, "Sliding window-based frequent pattern mining over data streams", Information Sciences, Vol. 179, 2009, pp. 3843-3865.

[16] H. F. Li and S. Y. Lee, "Mining frequent itemsets over data streams using efficient window sliding techniques", Expert Systems with Applications, Vol. 36, 2009, pp. 1466-1477.

[17] H. Li and H. Chen, "Mining non-derivable frequent itemsets over data stream", Data and Knowledge Engineering, Vol. 68, 2009, pp. 481-498.

[18] J. H. Chang and W. S. Lee, "estMax: Tracing maximal frequent itemsets instantly over online transactional data streams", IEEE Transactions on Knowledge and Data Engineering, Vol. 21, 2009, pp. 1418-1431.

[19] M. Deypir and M. H. Sadreddini, "EclatDS: An efficient sliding window based frequent pattern mining method for data streams", Intelligent Data Analysis, Vol. 15, 2011, pp. 571-587.

[20] Ahmed N. Albatineh and Magdalena Niewiadomska-Bugaj, "Correcting Jaccard and other similarity indices for chance agreement in cluster analysis", Adv Data Anal Classif, 5:179-200, DOI10.1007/s11634-011-0090-y, http://dx.doi.org/10.1007/s11634-011-0090-y, 2011.

[21] V.sidda Reddy, Dr T.V. Rao, and Dr A. Govardhan, "TIFIM: Tree based Incremental Frequent Itemset Mining over Streaming Data", International Journal of Computers & Technology, Vol 10, No 5, 2013.

[22] V.Sidda Reddy, Dr T.V. Rao, and Dr A. Govardhan, "Mining Frequent Itemsets (MFI) over Data streams: Variable Window Size (VWS) by Context Variation Analysis (CVA) of the streaming transactions", International Journal of Data Mining & Knowledge Management Process (IJDKP), Vol.4, No.4, July 2014.

[23] V.Sidda Reddy, M.Narendra and K.Helini, "Knowledge Discovery from Static Datasets to Evolving Data Streams and Challenges", International Journal of Computer Applications (IJCA), Volume 87, No.15, 2014, pp.22-25.