IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 14, 2019, accepted July 7, 2019, date of publication July 19, 2019, date of current version August 7, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2930004

Frequent Pattern Mining on Time and Location Aware Air Quality Data

APEKSHA AGGARWAL, (Member, IEEE), AND DURGA TOSHNIWAL, (Member, IEEE)
Department of Computer Science and Engineering, IIT Roorkee, Roorkee 247667, India

Corresponding author: Apeksha Aggarwal (aaggarwal@cs.iitr.ac.in)

This work was supported by the Ministry of Electronics and Information Technology (MeitY), Government of India.

ABSTRACT With the advent of big data era, enormous volumes of data are generated every second. Varied data processing algorithms and architectures have been proposed in the past to achieve better execution of data mining algorithms. One such algorithm is extracting most frequently occurring patterns from the transactional database. Dependency of transactions on time and location further makes frequent itemset mining task more complex. The present work targets to identify and extract the frequent patterns from such time and location-aware transactional data. Primarily, the spatio-temporal dependency of air quality data is leveraged to find out frequently co-occurring pollutants over several locations of Delhi, the capital city of India. Varied approaches have been proposed in the past to extract frequent patterns efficiently, but this work suggests a generalized approach that can be applied to any numeric spatio-temporal transactional data, including air quality data. Furthermore, a comprehensive description of the algorithm along with a sample running example on air quality dataset is shown in this work. A detailed experimental evaluation is carried out on the synthetically generated datasets, benchmark datasets, and real world datasets. Furthermore, a comparison with spatio-temporal apriori as well as the other state-of-the-art non-apriori-based algorithms is shown. Results suggest that the proposed algorithm outperformed the existing approaches in terms of execution time of algorithm and memory resources.

INDEX TERMS Air quality, data mining, frequent, itemset, spatio-temporal.

I. INTRODUCTION AND MOTIVATION

Web generates enormous volumes of heterogeneous data every second via sources such as social media, sensors, business enterprises etc. One such data that is focused upon in this work is air quality dataset, in addition to other transactional and synthetically generated datasets. There are various data mining methods that play a significant role in processing and analysing such data to extract useful information. Particularly association rule mining plays a critical role in applications such as market basket analysis, business [1] etc. However, due to the complexity of the real world big datasets [2], the need for efficient association rule mining algorithms for varied applications cannot be adjourned.

Prominent among them are the applications which generate spatio-temporal transactional data. Such datasets have a property that the information generated at one space and time behaves differently than the information generated at other space and time. Henceforth, to mine association rules among such databases, considering space-time information becomes obligatory. Spatio-temporal association rule mining is used for various applications in the past. Such as for mining disease [3], road networks [4] and mining bus Id card databases [5], marine environments [6], traffic [7] etc.

In this paper the main focus is to propose a generalize approach that can extract frequent patterns from spatio-temporal databases. In addition to mining spatio-temporal databases, our proposed approach calculates frequent spatio-temporal patterns at multiple levels of granularity. This granularity is explained in terms of spatial and temporal concept hierarchy levels. For example, time may be represented by as Year→Month→Day at different levels of concept hierarchy in descending order [8]. Furthermore, the present work is an extension of [9] in which a method to extract frequent items from categorical attributes is proposed. However, we extend this work for numeric attributes consisting of mutually exclusive items. Figure 1 shows the concept hierarchy for numeric attributes so as to make it suitable for spatio-temporal frequent itemset mining.

There were several shortcomings of previously proposed frequent itemset mining approaches. Primary of them includes repeated accesses of transactional database for

The associate editor coordinating the review of this manuscript and approving it for publication was Senthil Kumar.

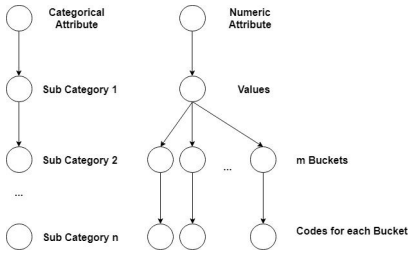


FIGURE 1. Concept hierarchy for categorical vs numeric attribute.

generating itemsets. This requires extremely large amount of memory storage as well as sharply increases the execution time for large databases. We propose hashing based spatio-temporal frequent itemset mining algorithm in this work which can be applied to varied spatio-temporal datasets including the air quality dataset. This work is an advantage over existing methods in terms of memory and execution time. Further this work applies the proposed algorithm on air quality data.

Air pollution levels are rising day by day. Extremely rising air pollution levels in the urban areas is one of the primary environmental concern of this era, which is to be addressed with the use of efficient techniques. Thus, the proposed spatio-temporal frequent itemset mining algorithm is applied on the air quality data of Delhi, the capital city of India, in order to identify frequently co-occurring air pollutant patterns. The air quality data consisted of parameters such as pollutant concentrations, location and time. Dependency of air quality data including concentrations of pollutants on spatial (latitude, longitude) and temporal (month, year) features makes it a suitable spatio-temporal transactional dataset. Henceforth, the primary aim of this work is to extract time and location aware frequent itemsets consisting of co-occurring pollutant concentrations using the proposed algorithm.

Our research contributions can be summarized as follows:

- The proposed research work is a hybrid version of CMS based approach and hash based searching techniques with efficient memory storage as well as execution time. The proposed Hash Based Spatio-Temporal (HBST) frequent itemset mining algorithm is computationally efficient and reduces overall execution time of the algorithm.
- Two step methodology is proposed to address the several aspects of mining spatio-temporal database i.e.
  - a Preprocessing step: to reduce the access time of database.
  - b Three phase approach: to reduce the number of accesses to the transactional database.
- The proposed work is an extension of [9] which was able to handle transactions with categorical attributes only. While the proposed work is applicable to any spatio-temporal databases containing numeric attributes, using discretization based on quantiles.
- Our approach is simple to implement and understand. Additionally, the proposed HBST algorithm ascertains reduced candidate generation.
- Finally, time and location aware frequent itemsets for air pollution data of Delhi are extracted in this work and a brief discussion about the reasons for co-occurrence of frequently occurring pollutants is given.

Rest of the paper is organized as follows: Section II presents the literature survey of existing frequent pattern mining algorithms in detail. Section III describes the methodology. Experiments and results are presented in section IV. Finally, conclusion is given in section V.

II. RELATED WORK

Atluri *et al.* [10] majorly classified spatio-temporal data mining techniques into six major categories: clustering, predictive learning, change detection, frequent pattern mining, anomaly detection, and relationship mining [8]. This work primarily focuses upon frequent pattern mining using association rule mining algorithms prevalant in the past.

Shaheen *et al.* [11] proposed a spatio-temporal association rule mining algorithm to identify the associations among different objects depending upon their context. Positive and negative frequent itemsets were identified in this work further utilizing context variable and spatial inputs of temporal series. Shao *et al.* [12] proposed ACAR, a supervised approach for software defect prediction based on atomic class-association rule mining. Further [13] suggested an algorithm to find frequent association patterns in data generated from smart devices and internet of things.

Chee *et al.* [14] classified frequent itemset mining algorithms into three categories, tree-based search algorithms (such as EclaT, TreeProjection etc.), pattern growth algorithms (such as FP- Growth, EXTRACT) and join-based algorithms (such as Apriori, DHP etc.).

Antonelli *et al.* [15] employed a fuzzy extension of FP-Growth algorithm for mining frequent patterns. However disadvantage of FP-Growth is its complex data structure and inefficiency on sparse datasets. To remove this, varied tree based data structures have been proposed in the past such as node list, node set etc. Aryabarzan *et al.* [16] proposed negFIN data structure which showed substantial reduction in total execution time required to mine frequent itemsets.

Wang *et al.* [17] suggested tree based temporal association rule mining algorithm. Liang *et al.* [18] used tree data structure to search frequent patterns. Turdukulov *et al.* [19] suggested frequent pattern discovery approach for moving flocks data. Szathmary [20] proposed Eclat- close, an extension of Eclat which is a vertical miner algorithm. Zhang *et al.* [21] proposed an extension of EclaT, a tree based algorithm. Primary limitation of these algorithms is high space complexity, specifically for larger transactions.

Qin *et al.* [22] utilized Apriori algorithm to project spatio-temporal effects of Particulate Matter in China. However, traditional algorithms such as EClaT, FP-Growth and Apriori were unable to adapt to spatio-temporal data environments.

One of the most popular algorithm for mining Spatio-Temporal association rules is Spatio-Temporal Apriori (STA) algorithm. STA is an extension of Apriori algorithm, which is

one of the prominent algorithms to mine association rules, in addition to FPGrowth, Eclat etc. [23]. Further several variants of Apriori had also been suggested in the past [24], which later became quite popular. Lin *et al.* [25] further utilized Apriori to find weighted frequent itemsets over uncertain databases.

In Spatio-Temporal Apriori [26], database was scanned again and again to generate candidate itemsets. This required high amounts of resources in terms of storage and execution times. Secondly, extremely large number of candidates generation in case of even less number of items, was further a problem. The algorithm proposed in [26] extensively reduced the number of accesses to the transactional database while extracting association rules.

In the present work the target was to reduce the total execution time of the algorithm. Total execution time constitutes the number of database accesses and time required to access the database. This work utilized calendar map schemas (CMS) [27], an additional schema for storing spatio-temporal information. CMS based approach reduce the number of database accesses, but limitation of existing CMS based approach was that repeated access of CMS may even increase the total execution time of the algorithm. Hence, hash based method [28], [29] have been employed in the present work, so that instead of accessing CMS repeatedly, hash keys are used. Notably, previous works utilized tree based data structures which are well known for their complex processing. We have not used tree based data structure in our work, instead we have utilized CMS data structures alongwith direct address hashing. We have used hashing in this work, because of its less time complexity. There are various hashing methods, but direct address hashing is utilized in order to avoid collision which can further increase the time complexity. Thus, hybrid of CMS based approach and hash based approach is proposed in this work, after removing the limitations of each of these approaches. Suggested arrangement not only have reduced the number of database access, but also reduced the time required to access the database.

Furthermore, several other research works to mine spatial and temporal association rules were given in the past [7], [30]–[32]. Park *et al.* [29] used hashing to mine association rules using dynamic hash tree. Winarko and Roddick [33] proposed ARMADA, an interval based temporal association rule mining algorithm. Further, [34] proposed STARminer algorithm an extension over [35] to extract large patterns in the database by considering only spatio-temporal association rules with high support and confidence. This algorithm aimed at finding sequences of object movements between regions. Notably, most of these approaches focused upon a specific type of transactional dataset. However, we have further proposed a generalized preprocessing algorithm based on quantile discretization so that our proposed method is suitable for any categorical or numerical dataset.

III. MATERIALS AND METHODS

A. STUDY AREA

Air quality (AQ) data of 5 locations of Delhi is considered as a case study in the present work. Delhi is the capital city of India which is among one of the most polluted cities of the world. Data from 5 air quality stations of Delhi are utilized for further analysis. Location areas are categorized into 3 types namely commercial, industrial and residential depending upon the land use patterns. Anand vihar and Shadipur locations are commercial areas, with most of the traffic intersections around the stations. Dwarka and RK Puram are the residential areas.

Punjabi Bagh location is mixed, with all the industrial, commercial and residential zones nearby. Figure 2 illustrates these locations annotated on maps. Three critical pollutants which are the primary contributors for degrading air quality in Delhi, are analysed namely, sulphur dioxide (SO2), nitrogen dioxide (NO2) and particulate matter 2.5 (PM2.5). Data readings are sampled at every 15 minute intervals over a period of 12 months i.e January 2016 to December 2016. Details about the data are given in Table 1.

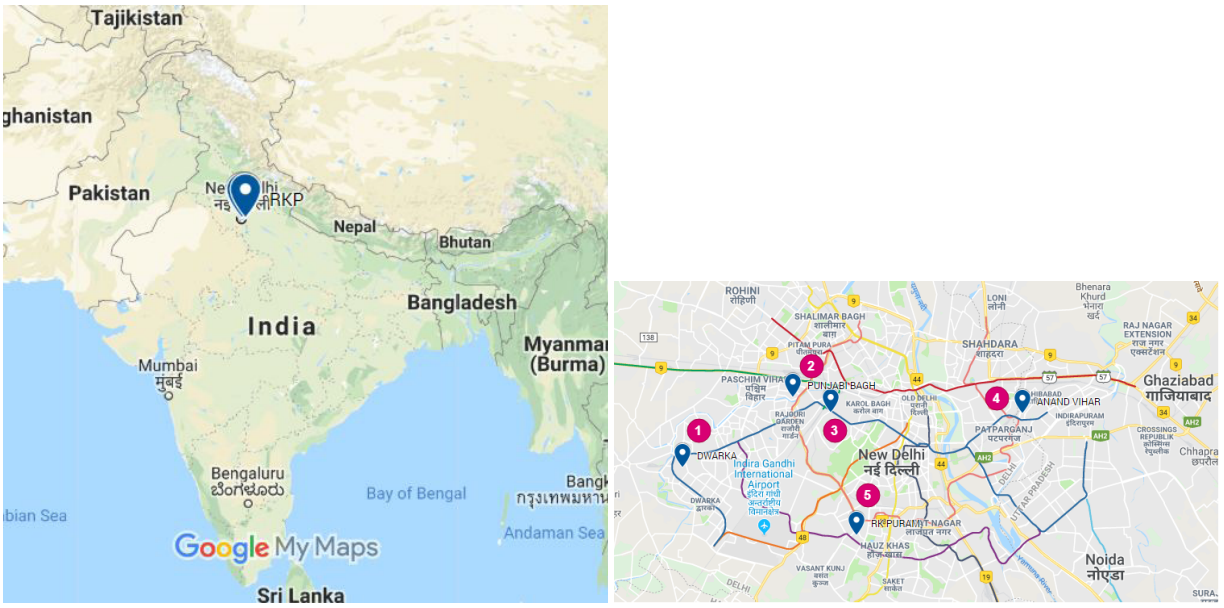
B. TERMS USED

Terminology adopted for the proposed method is explained as follows: Spatio-temporal database D contains a set of transactions, $trans_i$. Each transaction contains set of itemsets, I_k, where k is the number of items. Time and location based information is associated with each transaction for database D. Spatio-temporal schema or calendar map schema accommodate all the valid combinations of space and time out of all the possible combinations of users' transactions. For example, every market has a timing and no purchasing would occur outside that time window. That time window represents the temporal information, instead of 24 hours of the day. Definition for such schema is given as:

Definition 1: Calendar Map Schema (CMS) is the set of tuples formed by a set of either spatial or temporal granularities. CMS represents all the possible valid combinations of spatial and temporal information in which a transaction can occur.

TABLE 1. Data description.

S. no	Location	Latitude	Longitude	Transactions	Pollutant Analyzed	Area
1	Dwarka	28.61	77.03	84386	SO2, NO2, PM2.5	Residential
2	Punjabi Bagh	28.56	77.12	85833	SO2, NO2, PM2.5	Residential, Industrial and Commercial
3	Shadipur	28.65	77.15	13895	SO2, NO2, PM2.5	Commercial
4	Anand Vihar	28.65	77.32	82745	SO2, NO2, PM2.5	Commercial
5	RK Puram	28.56	77.17	25264	SO2, NO2, PM2.5	Residential



(a) Annotated map of India, depicting Delhi.

(b) Annotated map depicting all the spatial locations of Delhi.

FIGURE 2. Study area.

Figure 3 illustrates the structure of a CMS with spatial information represented by S_i and temporal information represented by T_i. Each of the transactions in Spatio-temporal database, D is associated with any of the tuple of the CMS. There are two types of tuples in a CMS [27] one containing basic spatio-temporal calendar map patterns (CMBs) and other containing '*', star based spatio-temporal calendar map patterns (CMPs). Note that '*' is a wild card symbol that denotes upper level of spatio-temporal information at concept hierarchy containing all the possible combinations at lower level. CMPs basically represents the transactions occurring in all the possible granularities of the valid space-time window. Besides CMS, several other definitions used in this work are defined as:

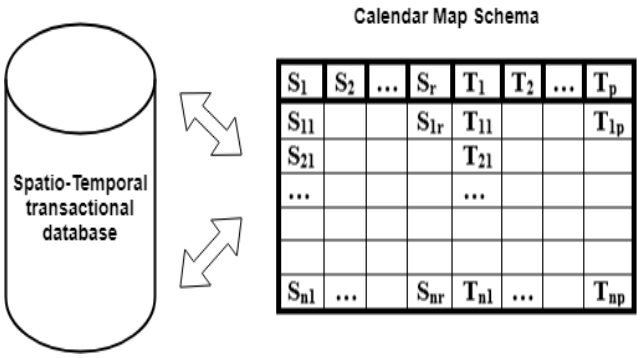


FIGURE 3. Figure illustrates the spatio-temporal transactional database, containing transactions and their corresponding information including a Calendar Map Schema representing all possible combinations of spatio-temporal information. Each spatial information and temporal information is represented by set of values i.e. $Domain(S_1) = \{S_{11}, S_{12}..S_{n1}\}$ and $Domain(T_1) = \{T_{11}, T_{12}..T_{n1}\}$ respectively. Each tuple of the CMS represents one valid combination of space-time information in which a transaction can occur.

Definition 2: Basic spatio-temporal Calendar Map pattern (CMB) is denoted by a single tuple in CMS representing space-time combination in which the occurring transaction contains no wild card entry symbol.

Definition 3: Star based spatio-temporal Calendar Map Pattern (CMP) is denoted by a single tuple in CMS representing space time combination in which the occurring transaction contains one or more wild card entry symbol.

Definition 4: A CMP tuple t_i is said to have covered another CMP tuple t_j if either each of the corresponding entries in t_i and t_j are equal or the corresponding entries in t_j contains a wild card entry symbol in t_i.

Definition 5: k-Calendar Map Pattern (k-CMP or k-star CMP) is the CMP containing k wild card entry symbols.

Definition 6: A pattern is called frequent if its support is greater than or equal to minimum support for a particular space-time granularity.

C. METHODOLOGY

This section presents the methodological steps to extract frequent patterns from air quality dataset consisting of pollutant concentrations and spatio-temporal parameters.

Step 1 (Data Preprocessing): Data are preprocessed in order to generate transactional spatio-temporal data sets on which the proposed algorithm can be employed. The air quality data set is cleaned, preprocessed and segmented into spatio-temporal partitions. Procedure to do so is illustrated in detail in Figure 4.

Step 2 (Quantiles Based Discretization to Generate Codes): Each of the attributes of PM2.5, SO2 and NO2 comprised of numeric attributes. To convert them into transactional data, algorithmic steps given in Figure 4 and algorithm 1 is utilized. Table 2 shows the summary statistics

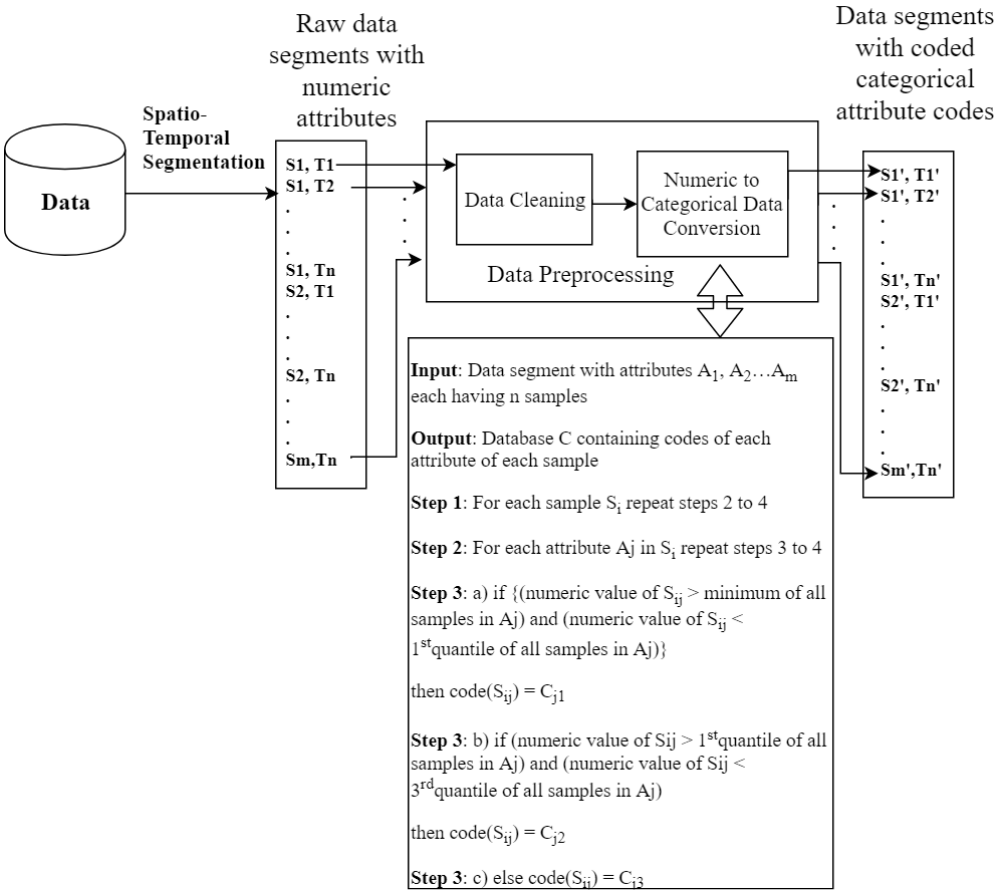


FIGURE 4. Data preprocessing steps.

TABLE 2. Summary statistics.

Loc	Anand Vihar			Shadipur			RK Puram			Dwarka			Punjabi Bagh		
	NO2	SO2	PM2.5	NO2	SO2	PM2.5	NO2	SO2	PM2.5	NO2	SO2	PM2.5	NO2	SO2	PM2.5
Min.	27.17	1.9	73	1.01	1.82	4.5	0.1	0.3	45	6	1.9	123.1	30.43	-0.97	51
1st Qu.	66.3	13.4	220.7	43.82	11.82	111.9	58.67	19.53	179	12.7	3.9	138.3	66.12	9.707	189.2
Mean	89.35	16.23	317	65.83	15.21	145.2	83.63	28.2	249	17.1	4.6	155.7	90.39	15.47	262.7
3rd Qu.	95.47	20.48	322.2	74.37	17.24	153.2	88.28	33.72	266.1	20.43	5.937	188.9	99.32	18.647	271.3
Max.	115.18	22.42	414	97.29	20.23	186.3	100.33	40.13	330	30.4	6.1	195.6	124.58	23	353.8

TABLE 3. Sample converted codes for Punjabi Bagh data.

NO2			SO2			PM2.5		
Greater Than or Equal To	Less Than	Code	Greater Than or Equal To	Less Than	Code	Greater Than or Equal To	Less Than	Code
30.43	66.12	NO21	-0.97	9.707	SO21	51	189.2	PM1
66.12	99.32	NO22	9.707	18.647	SO22	189.2	271.3	PM2
99.32	124.58	NO23	18.647	23	SO23	271.3	353.8	PM3

of the data including, minimum value of the attribute (min), 1st quantile (1st Qu.), mean, 3rd quantile (3rd Qu.) and maximum value (max). Table 3 shows sample code conversion for Punjabi Bagh data. Notably, this range is decided using Table 2 and steps given in algorithm 1.

Step 3 (CMS Generation): In this step, spatio temporal information were converted to calendar map schema. The number of spatio temporal partitions considered in this work are given in Figure 5. CMP (S1,*) denotes the transaction occurring at all the months over location S1. Similar can be stated for other locations and time.

Step 4 (HBST Algorithm): This step explains in detail the HBST algorithm. Aim of the proposed HBST frequent itemset mining algorithm is to identify frequent itemsets which holds enough number of identical spatio-temporal patterns using optimum resources. Existent disparity in search

Spatial Location	Temporal Information
Anand Vihar (S1)	January (T1)
Anand Vihar (S1)	February (T2)
...	...
Anand Vihar (S1)	December (T12)
Dwarka (S2)	January (T1)
Dwarka (S2)	February (T2)
...	...
Dwarka (S2)	December (T12)
Punjabi Bagh (S3)	January (T1)
Punjabi Bagh (S3)	February (T2)
...	...
Punjabi Bagh (S3)	December (T12)
RK Puram (S4)	January (T1)
RK Puram (S4)	February (T2)
...	...
RK Puram (S4)	December (T12)
Shadipur (S5)	January (T1)
Shadipur (S5)	February (T2)
...	...
Shadipur (S5)	December (T12)
S1	*
S2	*
...	*
S5	*
*	T1
*	T2
*	...
*	T12
*	*

Basic CMS

CMPs

FIGURE 5. Calendar map pattern for the given dataset.

Algorithm 1 Preprocessing Spatio-Temporal Transactions

Input:
Data segment with attributes A_1, A_2 ... A_m each having n samples

Output: Database C containing codes of each attribute of each sample

Description:

1: **for each** sample S_i repeat steps 2 to 5
2: **for each** attribute A_j in S_i repeat steps 3 to 4
3: a) **if** {(numeric value of S_{ij} > minimum of all samples in A_j) and (numeric value of S_{ij} < 1st quantile of all samples in A_j)}
code(S_{ij}) = C_{j1}
endif
3: b) **if** {(numeric value of S_{ij} > 1stquantile of all samples in A_j) and (numeric value of S_{ij} < 3rd quantile of all samples in A_j)}
code(S_{ij}) = C_{j2}
endif
3: c) **else** code(S_{ij}) = C_{j3}
4: **end for**
5: **end for**

techniques allowed us to choose hashing for quick memory access. Thus, curtailing execution time as well as memory storage. Proposed algorithm for spatio-temporal frequent itemset mining is given as per Algorithm 2 and 3 is also given in [9]. Before the application of algorithms, preprocessing step is taken. In this step, the whole calendar map schema, S containing basic and star based spatio-temporal patterns are concatenated with a hash id or hash address. This hash address is the concatenation of symbol specifying the n-star pattern and the unique auto-generated address. So hash addresses for n-star CMP is given as:

$$h = n + 1 \odot unique\_id \tag{1}$$

Definition 7: Hash function, f is defined as f ($\odot$) = h, which takes input value hash id and points to the location of corresponding specified address in memory.

Thus, whenever coverage of two calendar map patterns were compared with each other, instead of storing calendar map patterns with respect to frequent items, their hash addresses were stored. Figure 6 shows the sample example explaining the proposed algorithm. Firstly, 2-frequent itemsets were generated in first scan of the database. Direct generation of 2-itemsets is done so as to avoid the generation of redundant 1-itemsets [26]. Each partition specified the transactions denoted by similar spatio-temporal patterns. So, for each partitions representing basic CMPs, frequent 2-itemsets were generated. Counts of n-star patterns covering basic CMPs in each partitions were calculated and their corresponding hash values were stored. In the next step, k-candidate itemsets were generated using k-1 frequent itemsets and the hash addresses of corresponding frequent calendar map patterns were updated. Itemsets with support less than user specified minimum support threshold, *min_sup*

Algorithm 2 Hash Based Spatio-Temporal Frequent Itemsets

Input: D: Transaction database, *CMBs*: Basic spatio-temporal calendar map patterns, *CMPs*: n-star spatio-temporal calendar map patterns;
Min_sup: the minimum support threshold

Output: frequent itemsets and frequent Spatio-Temporal patterns

Description:

1: **for each** partition $p \in P_i$ {
   // partition P_i is a set of transactions belonging to same tuple in CMB
2: $L_2 =$ find_frequent_2-itemsets (P_i, s_1); //find 2-frequent itemsets in all basic space-time intervals
3: **end for**
4: **for each** $l_2 \in L_2$
5: $l_2.h_i^1 = f(1\_starCMPs(l_2))$ such that $CMPs(l_2) \subseteq 1\_starCMPs(l_2)$ //$L_2.H^1$ is a set of hash values of all 1_star CMPs containing l_2
6: Update $l_2.h_i^1.count$ //using hash updation procedure
7: **for**$(i = 2; i = n; i++)$ {
8: if $CMPs(L_1) \subseteq i - starCMPs$\{
9: $l_2.h_j^i = f((i-1)\_starCMPs(l_2))$
10: Update $l_2.h_j^i.count$ //using hash updation procedure
11: **end for**
12: **end for**
13: $C_2 = L_2$
14: **for**(k $= 3$; $L_{k-1} = \varphi$; $k++$) //for each candidate k-itemset C_k
15: Calculate $H_{I_q^j}$ for each item I_q^j in C_k
16: Update $c_k.H.count$ //using hash updation procedure
17: **end for**
18: **return** return $C = \bigcup_k C_k$ and $v.C \bigcup_k h.C_k \,|C_k.count > min\_sup$;

were removed. In step 3, counts on n star CMPs were updated using n-1 star CMPs. Lastly, step 2 and 3 were repeated until no more frequent itemsets can be generated. Algorithm 2 and 3 explains the procedure for generating frequent itemsets.

Definition 8: Support of an item is defined as the frequency of occurrence of an item in a partition representing the similar space-time information.

Figure 6 shows a spatio-temporal database with a set of transactions D and each transaction is associated with a spatial and temporal information, S and T. CMS represents all the possible valid combinations of spatial and temporal information. Spatial granularity, S can contain values S1 and S2 and Temporal granularity, T can contain values T1 and T2 including a wild card entry symbol i.e. $Domain(S) = \{S_1, S_2, \ldots, S_m, *\}$ and $Domain(T) = \{T_1, T_2, \ldots, T_n, *\}$. Wild card entry symbol * in S represents that a transaction is occurring in all the spatial locations. Similar can be stated for T.

In step 1 of figure 6, hash ids were inserted which were a combination of (n+1) value and a unique id. For example

Algorithm 3 Hash Updation Procedure to Generate Frequent Itemsets

Input: L_k: Set of k-itemsets, *CMBs*: Basic spatio-temporal calendar map patterns, *CMPs*: n-star spatio-temporal calendar map patterns;

Output: updated hash counts in spatio-temporal patterns

Description:

1: **if** $\{l_k \notin L_k\}$
2: insert l_k in L_k and include the hash value of corresponding n-star CMPs in $l_k.H^n$.
3: $l_k.h_i^n.$count $= 1$.
4: **end if**
5: **if** $\{l_k \in L_k$ and $l_k.h_i^n \notin l_k.H^n\}$
6: include the hash value of corresponding n-star CMPs in $l_k.H^n$.
7: $l_k.h_i^n.$count $= 1$.
8: **end if**
9: **if** $\{l_k \in L_k$ and $l_k.h_i^n \in l_k.H^n\}$
10: $l_k.h_i^n.$count++.
11: **end if**
12: **return** $l_k.H^n$ // set of all the hash values of n-star candidate calendar-map patterns of l_k

all the 1* CMPs were a concatenation of '2' and an identifier id. So, CMPs with the hash ids 21 and 22 represents 1*CMPs as per the figure. In step 2, database is partitioned according to the transactions having the same spatio-temporal information. 2-frequent itemsets were generated in each of these partitions corresponding to 1 basic CMP. Figure shows several partitions representing the basic CMPs in which transaction occurs. In step 3, the counts of 2-frequent itemsets in each partition were updated with respect to the corresponding 1*CMPs in which they occur. Alongwith the counts, the hash ids were also updated. Procedure for updation is shown in Algorithm 3. This process is repeated in step 4 for 2* CMPs. In step 5, 3-itemsets were generated using previously generated 2-itemsets and their frequency counts were updated using Algorithm 3. And this process is repeated for all the k-itemsets. After completion of these steps, output is compiled by the union of all the k-itemsets and their support counts generated at each step. All the itemsets whose minimum support is greater than or equal to *min_sup* were considered as frequent. Formula for support is given as:

$$Support(X) = \frac{Support\ count\ of\ item\ X}{Number\ of\ Transactions} \tag{2}$$

IV. EXPERIMENTS AND RESULTS

A. DATA AND TOOLS

To evaluate the performance of our algorithm, experiments are performed on 3 kinds of datasets. First is benchmark datasets obtained from FIMI repository [36]. Three benchmark datasets are utilized whose descriptions are given in Table 4. Second is synthetically generated dataset which is also available online [36]. For generating random transactions, random simulation method is used assuming

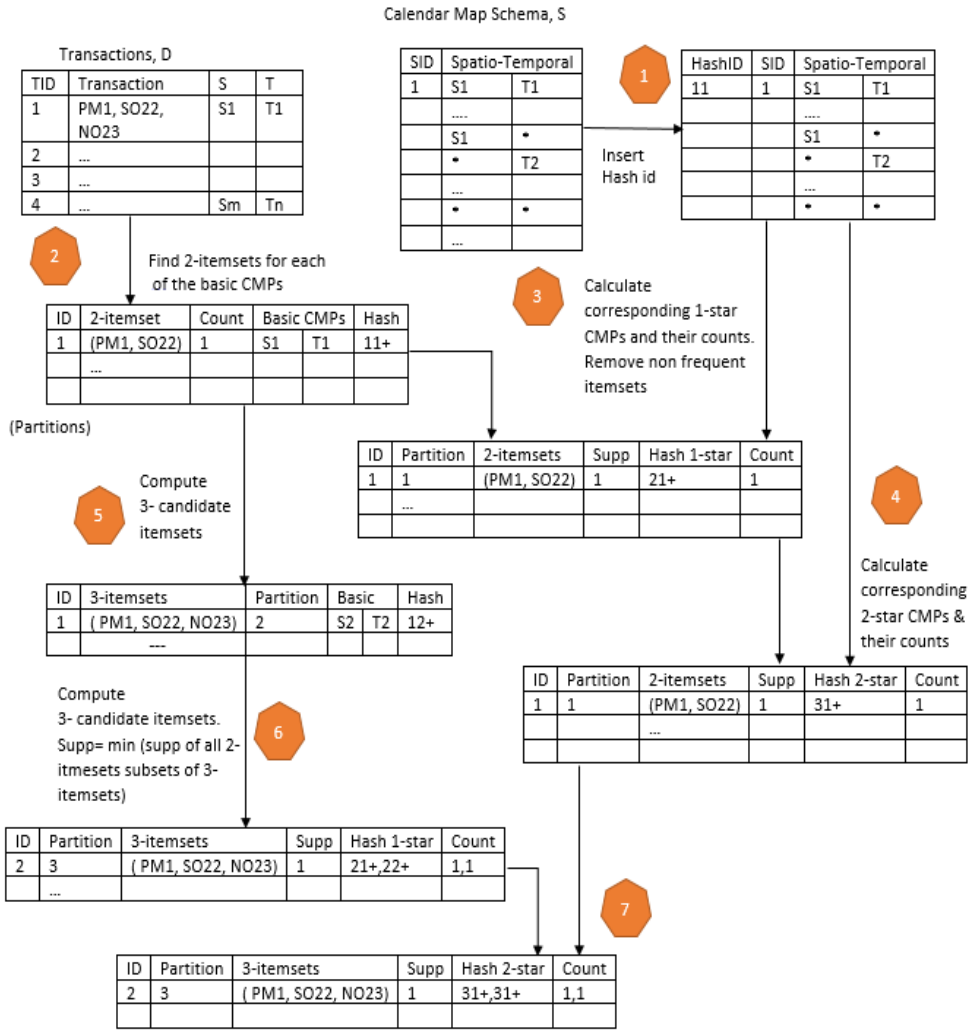


FIGURE 6. Steps depicting Hash Based Spatio-Temporal (HBST) frequent itemset generation procedure for air quality data. Note that '+' indicates the hash address, where hash function points to.

TABLE 4. Datasets description.

Dataset Name	Number of Transactions	Number of Items
Accidents1	340,183	468
Chess	3,196	75
Mushroom	8,124	119
T10I4D100K	98,487	949

all the transactions occur independently irrespective of the correlations among them. Third, we have applied the proposed algorithm on air quality dataset of India, comprising of numerical attributes and the results are obtained. Description of air quality dataset is given in detail in section III.

For generating spatio-temporal partitions over the available benchmark datasets, R programming tool Version 1.1.453 is used. For implementation of algorithms, we have used Scientific PYthon Development EnviRonment 3.2.8 with Python 3.6.5 64bits, Qt 5.9.4, PyQt5 5.9.2. We have conducted all our experiments on Windows 10, with Intel(R) core TM i7-4790 CPU @3.60 GHz processor and 32.0 GB memory.

B. PERFORMANCE EVALUATION

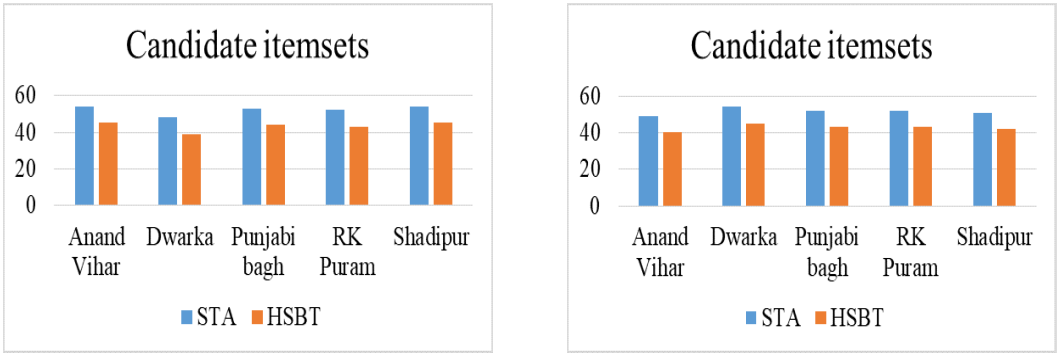
Two groups of experiments are performed to evaluate the performance of the proposed algorithm. First is the comparison of our proposed HBST algorithm against non-Apriori based state-of-the-art algorithms such as FPGrowth and Eclat [14]. Second one is the comparison of our algorithm with Spatio-Temporal Apriori (STA) algorithm given in [9], which followed the similar structure as HBST.

We ran our proposed algorithm and the state-of-the-arts on all the datasets for varying support values, i.e. 10%, 20%, 30% and 50%. Evaluation is done over various parameters namely:

- Effect of the size of transactional database
- Execution time for k-Frequent Itemsets Generation
- Memory Usage

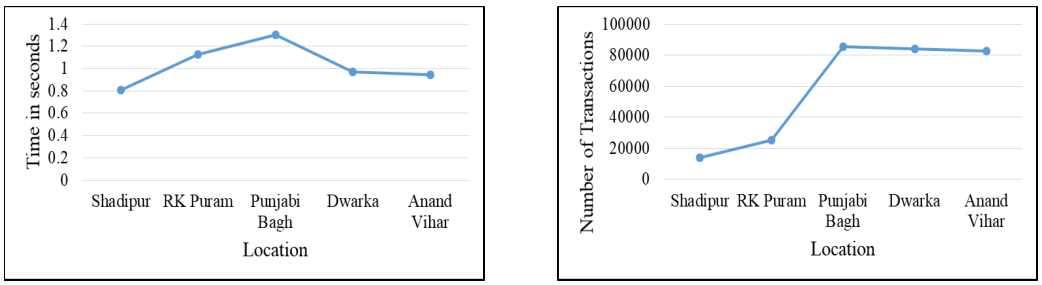
C. RESULTS 1: MEMORY USAGE

Memory required at each step of the algorithm is reduced by using hashing addresses or ids in place of storing the whole CMPs repeatedly. This saved the memory storage as well as reduce the execution time of the algorithm. Further, for



(a) Month of May. (b) Month of October.

FIGURE 7. Candidate k-itemset generated for basic CMP.



(a) Number of Transactions. (b) Execution time.

FIGURE 8. Time taken to generate candidate k-itemset generated for basic CMB with varying number of transactions.

Apriori technique ${}^nC_k$ number of candidate itemsets were generated, where n is the number of items and k is the itemset. So, for our datasets, number of 2-itemsets is ${}^nC_2$ and this number kept on growing exponentially for upto k-itemsets. Similarly, for Spatio-Temporal Apriori, number of candidates generated at each step grow exponentially. But in our proposed approach, instead of scanning the database again and again, candidates were generated from previously large 2-frequent itemsets only. Thus, reduced number of candidates generation, further curtailed the time and memory resources. Figure 7 shows the number of candidate k-itemsets generated for these two algorithms for the month of May and October over all the locations.

D. RESULTS 2: EFFECT OF THE SIZE OF TRANSACTIONAL DATABASE

Figure 8 depicts the time required to generate basic CMS spatio-temporal frequent itemsets on the HBST algorithm, performed over given partitions. Note that the maximum k-itemset size on all the CMB data partitions are the same, however the data partition size varies. Aim is to evaluate the performance of our algorithm over datasets of varying sizes. For this experiment, we have partitioned data of different locations into different transactions.

Figure shows the performance of HBST algorithm is not much affected by the size of the database partition. Figure 8a shows even with the abrupt increase in size of data, there is not much increase in execution time of the algorithm, illustrated in Figure 8b.

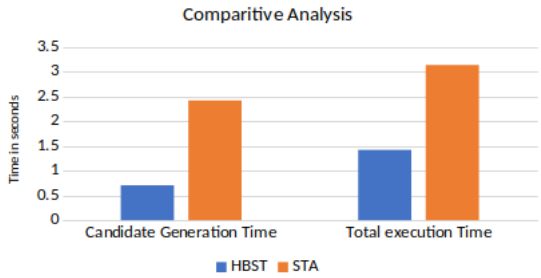


FIGURE 9. Comparison between STA and HBST.

The reason for this might be because of reduced number of transactional database accesses, the effect of large sizes of the databases masked out. Furthermore, our proposed HBST algorithm reduced the number of transactional database accesses in step 1 of the algorithm by calculating the 2-frequent itemsets directly. This further reduced the generation of 1-itemsets which were not frequent. Secondly, after computing 2-frequent itemsets, our proposed algorithm calculated support counts for each partition, by using CMS only, hence the transactional database was not required to be loaded into memory again and again. Subsequently, 3-itemsets were generated using 2-itemsets and so on.

E. RESULTS 3: EXECUTION TIME

Total execution time of an algorithm comprised of the number of accesses to the transactional database and generated candidates plus the time required to access the intermediate storage variables containing candidates generated including the transactional database. For this experiment, whole dataset

TABLE 5. Sample candidate itemset generation for Anand Vihar, over the month of January.

k value	Sample candidate k-itemsets and support count
2	{'pm2', 'so22'}: 361, {'no22', 'so22'}: 413, {'pm2', 'no22'}: 341, {'so21', 'pm1'}: 584, {'no22', 'so21'}: 293, {'no22', 'pm1'}: 394, {'no21', 'so21'}: 705, {'no21', 'pm1'}: 582, {'pm2', 'no21'}: 384, {'pm2', 'so21'}: 363, {'no21', 'so23'}: 16, {'pm3', 'so23'}: 39, {'no21', 'pm3'}: 51
3	{'pm2', 'no22', 'so22'}: 163, {'no22', 'so21', 'pm1'}: 140, {'no21', 'so21', 'pm1'}: 437, {'pm2', 'no21', 'so21'}: 232, {'no21', 'pm3', 'so23'}: 4

is considered containing 60 basic CMB patterns and 18 1-star and 2-star partitions. All the steps of the proposed algorithm were executed as per Figure 6 and stepwise total execution time of the algorithm is calculated. In Figure 9, a comparison between total execution time calculated for two algorithms STA and HBST is given which suggests that our algorithm drastically improves this candidate generation time and overall execution time of the algorithm.

F. RESULTS 4: COMPARISON OVER NON-APRIORI BASED METHODS

This section presents a comparison of proposed HBST algorithm with other non-Apriori based methods such as FPGrowth [15] and Eclat [21], which are one of the most popular and well established algorithms for frequent itemset mining. Results over 3 types of datasets are presented in this section. Firstly, our algorithm is employed over air quality dataset of India and results were evaluated. Figure 10 illustrates the stepwise time taken by the algorithm for generating frequent itemsets (FIs). Figures suggests a remarkable improvement in execution times of our proposed algorithms over state-of-the-arts. Secondly, we have employed our algorithm over a few benchmark datasets named 'Accidents1', 'Chess' and 'Mushroom' dataset and a synthetically generated dataset named 'T10I4D100K'. Note that all of these datasets are not spatio-temporal datasets, so we have partitioned these into different spatial and temporal partitions randomly. Sampling without replacement method is used to generate a set of spatio-temporal partitions. Results over all these datasets are illustrated in Figure 11. Further, figure illustrates the significant improvement of HBST algorithm over existing state-of-the-arts in terms of total execution times of algorithm, calculated for varying support values. Note that 'T10I4D100K' dataset has the maximum number of items, but still the results suggests an improvement in our algorithm over existing ones in terms of total execution time of the algorithm.

G. RESULTS 5: FREQUENT ITEMSETS

Table 5 shows the sample candidate itemsets generated for location S1 and time T1 over different support counts, for the air quality dataset of India. Similarly, Table 6 and 7 shows several CMBs and CMP patterns generated. Note that formula for calculating support is used as per equation given in previous section. Note that codes utilized in these tables are explained in detail in *Step 2* and *Step 3* of section III. Table 6 shows several 2-frequent itemsets and 3-frequent itemsets for several locations and on several time periods respectively for air quality dataset of India. Table 7 shows several 3-frequent spatio-temporal itemsets generated over air quality dataset of India, depicting co-occurring pollutants at varied space-time granularities.

H. DISCUSSION

Table 7 showed 3-itemsets generated for 2 star and 1 star CMPs. The extracted frequent itemsets are taken for manual sampling and analyses. Since, we have no other way

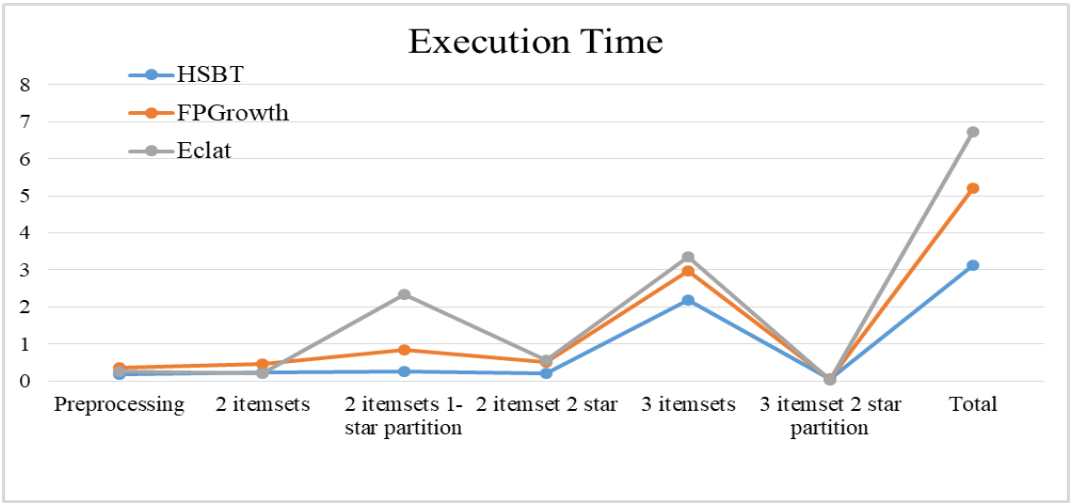


FIGURE 10. Runtime comparison against state-of-the-arts. X axis depicts various steps of the algorithm and Y axis depicts the execution time in seconds.

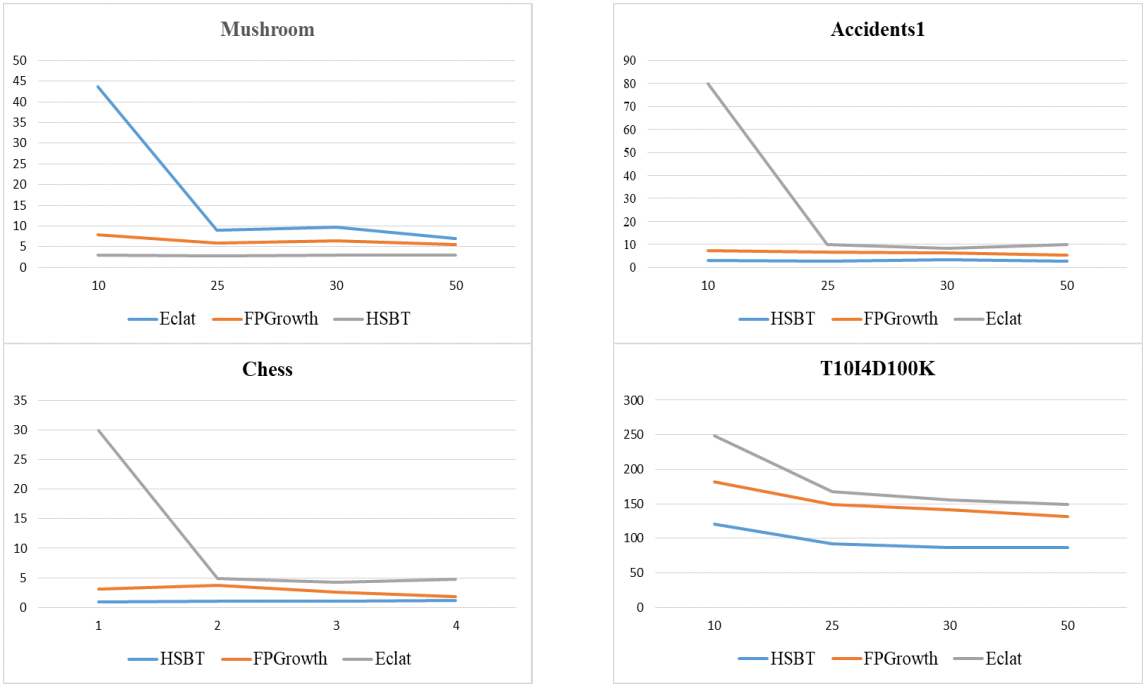


FIGURE 11. Runtime comparison against benchmark datasets over varying support values. X axis depicts minimum support in percentage and Y axis depicts the execution time in seconds.

TABLE 6. Frequent itemsets for several basic CMBs.

CMB	Location, Time	2- Frequent Itemsets and 3- Frequent Itemsets Generated
S2,T5	Dwarka, May	{'pm1', 'no22'}, {'so21', 'no22'}, {'so21', 'pm2'}, {'pm2', 'no23'}, {'no22', 'pm2'}, {'pm2', 'so22'}, {'pm2', 'no21'}, {'so21', 'pm1'}, {'so21', 'no21'}, {'no21', 'pm1'}, {'no22', 'pm1'}, {'no22', 'so22'}, {'so21', 'no22', 'pm1'}, {'so21', 'no21', 'pm1'}
S2,T7	Dwarka, July	{'no22', 'pm2'}, {'no21', 'pm1'}, {'so21', 'no22'}, {'so21', 'pm1'}, {'pm2', 'so22'}, {'no22', 'pm1'}, {'no22', 'so22'}, {'pm2', 'no21'}, {'so21', 'no21'}, {'so21', 'pm2'}, {'so21', 'no21', 'pm1'}, {'so21', 'pm2', 'no21'}
S2,T9	Dwarka, September	{'no22', 'so22'}, {'pm2', 'no21'}, {'pm2', 'so22'}, {'so21', 'pm2'}, {'no22', 'pm2'}, {'no22', 'pm1'}, {'so21', 'no22'}, {'no21', 'pm1'}, {'so21', 'no21'}, {'so21', 'pm1'}, {'so21', 'no21', 'pm1'}
S2,T11	Dwarka, November	{'no22', 'so22'}, {'no22', 'pm1'}, {'so21', 'no21'}, {'no21', 'pm1'}, {'so21', 'pm1'}, {'so21', 'no22'}, {'so21', 'pm2'}, {'pm2', 'no21'}, {'pm2', 'so22'}, {'so21', 'no22', 'pm1'}, {'so21', 'no21', 'pm1'}

TABLE 7. 3-frequent itemsets for 1-star and 2-star CMPs.

CMP	3- Frequent Itemsets Generated
S2,*	{'so21', 'no21', 'pm1'}
*, T5	{'so21', 'no22', 'pm1'}, {'so21', 'no21', 'pm1'}
*, T10	{'pm2', 'no21', 'so22'}
S3,*	{'so21', 'no22', 'pm1'},{'no22', 'so23', 'pm2'}, {'pm2', 'no23', 'so22'},{'no22', 'pm2', 'so22'}, {'so21', 'pm2', 'no21'}, {'so21', 'no21', 'pm1'}, {'pm2', 'no21', 'so22'}
S4,*	{'so21', 'pm2', 'no21'}, {'so22', 'no22', 'pm3'}, {'no22', 'pm2', 'so22'}, {'pm2', 'no23', 'so22'}, {'no22', 'so23', 'pm2'}
,	{'no22', 'pm2', 'so22'},{'so21', 'no21', 'pm1'}, {'pm2', 'no23', 'so22'}, {'so21', 'pm2', 'no21'}

to characterize the truth about the extracted results, we utilized literature and print media sources to verify the ground truth about the results. Results suggested the co-occurrence of low PM2.5, low NO2 and low SO2 {'so21', 'no21', 'pm1'} for all the regions and all the months of a year as frequent itemset which is quite obvious. However, the results also suggested the co-occurrence of high NO2 with average PM2.5 and average SO2 {'pm2', 'no23', 'so22'}. A detailed

TABLE 8. Analysis of pollution sources at various locations of Delhi.

S. no	Location	High Pollution Sources
1	Anand Vihar	Domestic activities, vehicular movement, roadside eatouts and open burning of leaves and solid waste.
2	Punjabi Bagh	Punjabi Bagh is mixed area but near to most of the industrial areas.
3	Shadipur	Traffic intersections, nearby industries and other commercial centers.
4	Dwarka	Dwarka is residential area and air quality can be affected mainly by domestic activities such as cooking, generator sets for power backup.
5	RK Puram	RK Puram is also residential area with similar pollution sources.

analysis of reasons for high concentration of NO2 in several parts of India is given in [37]. Several studies suggested hotspots for increasing SO2 and NO2 [38] in several locations of India along with PM2.5 justifying itemset {'pm2', 'no22', 'so22'} to be frequent with medium concentrations of each of these pollutants. However, these results are specific to a time and location [39], [40]. For example, winter months suffer from a great pollution than other months justifying the presence of {'pm2', 'no21', 'so22'} in (*, T10). Similarly, locations with more industrial areas, traffic intersections, thermal power plants etc. suffer from high pollution levels usually. Location of Punjabi bagh (S3,*) is surrounded by industrial areas from various sides [37]. Hence, the presence of 'so23' and 'no23' in their itemsets is more frequent than other spatial locations. Table 8 illustrates the possible reasons for the findings. Finally, frequent itemsets in (*,*) CMP for all locations and all the months suggests that presence of low values of pollutants is correlated at all the places, however medium values of NO2 and SO2 are accompanied by medium or high values of PM2.5. The important point that has been noted further in the above results is for most of the locations, there have been very few instances of co-occurrence of high pollutants with each other. Possible reason for this could be the fact that all the different pollutants are emitted from the different sources which might not be overlapping. Similarly, the locations with source of one pollutant might not be the source for other pollutants. Thus, we have identified the time and location based dependency of pollutants in the air of Delhi.

V. CONCLUSION

With the commencement of associated location and temporal information along with the transactions, efficient algorithms are required for extracting frequent itemsets from such databases. Existing algorithms require huge amounts of resources in terms of execution time for candidate generation, number of accesses to the database as well as the time required to access large spatio-temporal databases. In this work, the number of database accesses are reduced by using CMS. But issue with existing approach is that the repeated access of CMS may even reduce the execution time of the algorithm. So, we suggested the use of direct address hashing. Furthermore, this work proposed spatio-temporal frequent itemsets mining algorithm to extract frequent items at multiple levels of granularities. Direct address based hashing technique is used, so as to optimize collisions as well as the execution time of the algorithm. Experiments are performed over benchmark datasets, synthetically generated datasets as well as real world numeric dataset containing concentrations of several pollutants in air of Delhi. Additionally, a comparison with the already existing apriori based algorithms such as STA is given. Results suggested that the time required to execute the different steps of the these algorithms was far less in proposed HBST than STA. Secondly, even for large number of items, our algorithm performed drastically better than STA in terms of total execution time of the algorithm. Thirdly, memory taken at each step of the algorithm is reduced when HBST algorithm is applied on the datasets. Furthermore, the results are compared over various other non apriori based methods such as FPGrowth and Eclat. Results suggested that our algorithm outperformed the existing algorithms in terms of memory and execution times. Further frequent co-occurring pollutant patterns are extracted for air quality data of Delhi using the proposed algorithm and a detailed discussion over the results is provided. In future we plan to implement this work for various other web datasets.

ACKNOWLEDGMENT

The authors would like to thank the institute for providing infrastructure and support from *Visvesvaraya PhD Scheme for Electronics and IT*, under Ministry of Electronics and Information Technology, Government of India, to carry out this research. Authors would also like to thank IEEE for the waiver.

REFERENCES

[1] J. Han, H. Cheng, D. Xin, and X. Yan, "Frequent pattern mining: Current status and future directions," *Data Mining Knowl. Discovery*, vol. 15, no. 1, pp. 55–86, Aug. 2007.
[2] X. Yao, "Research issues in spatio-temporal data mining," in *Proc. Workshop Geospatial Visualizat. Knowl. Discovery, Univ. Consortium Geographic Inf. Sci., Virginia*, 2003, pp. 1–6.
[3] V. Raheja and K. Rajan, "Comparative study of association rule mining and MiSTIC in extracting spatio-temporal disease occurrences patterns," in *Proc. IEEE 12th Int. Conf. Data Mining Workshops*, Dec. 2012, pp. 813–820.
[4] W. Yu, "Spatial co-location pattern mining for location-based services in road networks," *Expert Syst. Appl.*, vol. 46, pp. 324–335, Mar. 2016.
[5] X.-L. Zhao and W.-X. Xu, "Mining spatio-temporal association rules in bus IC card databases," in *Proc. 2nd Int. Conf. Power Electron. Intell. Transp. Syst. (PEITS)*, vol. 1, Dec. 2009, pp. 125–128.
[6] C. Xue, S. Wanjiao, Q. Lijuan, D. Qing, and W. Xiaoyang, "A spatiotemporal mining framework for abnormal association patterns in marine environments with a time series of remote sensing images," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 38, pp. 105–114, Jun. 2015.

[7] H. Nguyen, W. Liu, and F. Chen, "Discovering congestion propagation patterns in spatio-temporal traffic data," *IEEE Trans. Big Data*, vol. 3, no. 2, pp. 169–180, Jun. 2016.
[8] J. Han, J. Pei, and M. Kamber, *Data Mining. Concepts and Techniques*. Amsterdam, The Netherlands: Elsevier, 2011.
[9] A. Aggarwal and D. Toshniwal, "Spatio-temporal frequent itemset mining on Web data," in *Proc. IEEE Int. Conf. Data Mining Workshops (ICDMW)*, Nov. 2018, pp. 1160–1165.
[10] G. Atluri, A. Karpatne, and V. Kumar, "Spatio-temporal data mining: A survey of problems and methods," *ACM Comput. Surv.*, vol. 51, p. 83, Sep. 2018.
[11] M. Shaheen, M. Shahbaz, and A. Guergachi, "Context based positive and negative spatio-temporal association rule mining," *Knowl.-Based Syst.*, vol. 37, pp. 261–273, Jan. 2013.
[12] Y. Shao, B. Liu, S. Wang, and G. Li, "A novel software defect prediction based on atomic class-association rule mining," *Expert Syst. Appl.*, vol. 114, pp. 237–254, Dec. 2018.
[13] S. A. Aljawarneh, R. Vangipuram, V. K. Puligadda, and J. Vinjamuri, "G-SPAMINE: An approach to discover temporal association patterns and trends in Internet of Things," *Future Gener. Comput. Syst.*, vol. 74, pp. 430–443, Sep. 2017.
[14] C.-H. Chee, J. Jaafar, I. A. Aziz, M. H. Hasan, and W. Yeoh, "Algorithms for frequent itemset mining: A literature review," *Artif. Intell. Rev.*, vol. 2, pp. 1–19, Mar. 2018.
[15] M. Antonelli, P. Ducange, F. Marcelloni, and A. Segatori, "A novel associative classification model based on a fuzzy frequent pattern mining algorithm," *Expert Syst. Appl.*, vol. 42, no. 4, pp. 2086–2097, Mar. 2015.
[16] N. Aryabarzan, B. Minaei-Bidgoli, and M. Teshnehlab, "negFIN: An efficient algorithm for fast mining frequent itemsets," *Expert Syst. Appl.*, vol. 105, pp. 129–143, Sep. 2018.
[17] L. Wang, J. Meng, P. Xu, and K. Peng, "Mining temporal association rules with frequent itemsets tree," *Appl. Soft Comput.*, vol. 62, pp. 817–829, Jan. 2018.
[18] P. Liang, J. F. Roddick, and D. de Vries, "Searching frequent pattern and prefix trees for higher order rules," *Data Mining Anal.*, vol. 13, p. 129, May 2013.
[19] U. Turdukulov, A. O. C. Romero, O. Huisman, and V. Retsios, "Visual mining of moving flock patterns in large spatio-temporal data sets using a frequent pattern approach," *Int. J. Geographical Inf. Sci.*, vol. 28, no. 10, pp. 2013–2029, Apr. 2014.
[20] L. Szathmary, "Finding frequent closed itemsets with an extended version of the eclat algorithm," in *Annales Mathematicae et Informaticae*, vol. 48, pp. 75–82, Jan. 2018.
[21] C. Zhang, P. Tian, X. Zhang, Q. Liao, Z. L. Jiang, and X. Wang, "HashEclat: An efficient frequent itemset algorithm," *Int. J. Mach. Learn. Cybern.*, vol. 1, pp. 1–14, Jan. 2019.
[22] S. Qin, F. Liu, C. Wang, Y. Song, and J. Qu, "Spatial-temporal analysis and projection of extreme particulate matter (PM_{10} and $PM_{2.5}$) levels using association rules: A case study of the Jing-Jin-Ji region, China," *Atmos. Environ.*, vol. 120, pp. 339–350, Nov. 2015.
[23] J. Hipp, U. Güntzer, and G. Nakhaeizadeh, "Algorithms for association rule mining—A general survey and comparison," *ACM SIGKDD Explor. Newslett.*, vol. 2, no. 1, pp. 58–64, Jun. 2000.
[24] Y. PENG and Y. XIONG, "Study on optimization of aprioritid algorithm for mining association rules," *Comput. Eng.*, vol. 5, p. 019, Mar. 2006.
[25] J. C.-W. Lin, W. Gan, pp. Fournier–Viger, T.-P. Hong, and V. S. Tseng, "Weighted frequent itemset mining over uncertain databases," *Appl. Intell.*, vol. 44, no. 1, pp. 232–250, Apr. 2016.
[26] E. M. Lee and K. C. Chan, "Discovering association patterns in large spatio-temporal databases," in *Proc. 6th IEEE Int. Conf. Data Mining*, Dec. 2006, pp. 349–354.
[27] Y. Li, P. Ning, X. S. Wang, and S. Jajodia, "Discovering calendar-based temporal association rules," *Data Knowl. Eng.*, vol. 44, no. 2, pp. 193–218, 2003.
[28] J. S. Park, M.-S. Chen, and P. S. Yu, "An effective hash-based algorithm for mining association rules," *ACM SIGMOD Rec.*, vol. 24, no. 2, pp. 175–186, May 1995.
[29] J. S. Park, M.-S. Chen, and P. S. Yu, "Using a hash-based method with transaction trimming for mining association rules," *IEEE Trans. Knowl. Data Eng.*, vol. 9, no. 5, pp. 813–825, Sep. 1997.
[30] V. Radhakrishna, S. A. Aljawarneh, P. V. Kumar, and K.-K. R. Choo, "A novel fuzzy gaussian-based dissimilarity measure for discovering similarity temporal association patterns," *Soft Comput.*, vol. 22, no. 6, pp. 1903–1919, Mar. 2018.
[31] L. Xiong, X. Liu, D. Guo, and Z. Hu, "Access patterns mining from massive spatio-temporal data in a smart city," *Cluster Comput.*, vol. 1, pp. 1–11, Jan. 2018.
[32] J. F. Roddick and M. Spiliopoulou, "An updated bibliography of temporal, spatial, and spatio-temporal data mining research," *ACM SIGKDD Explor. Newslett.*, vol. 1, no. 1, pp. 34–38, Mar. 1999.
[33] E. Winarko and J. F. Roddick, "Armada–an algorithm for discovering richer relative temporal association rules from interval-based data," *Data Knowl. Eng.*, vol. 63, no. 1, pp. 76–90, 2007.
[34] F. Verhein, "k-STARs: Sequences of spatio-temporal association rules," in *Proc. 6th IEEE Int. Conf. Data Mining*, Dec. 2006, pp. 387–394.
[35] F. Verhein and S. Chawla, "Mining spatio-temporal association rules, sources, sinks, stationary regions and thoroughfares in object mobility databases," in *Proc. Int. Conf. Database Syst. Adv. Appl.*, 2006, pp. 187–201.
[36] (2019). *Frequent Itemset Mining Implementations Repository*. Accessed: Jun. 9, 2019. [Online]. Available: http://fimi.uantwerpen.be/
[37] A. Aggarwal and D. Toshniwal, "Detection of anomalous nitrogen dioxide (NO_2) concentration in urban air of India using proximity and clustering methods," *J. Air Waste Manage. Assoc.*, vol. 69, no. 7, pp. 805–822, 2018.
[38] T. Hindu. (2018). *Thermal Power Plants Leading to Spike in SO_2, NO_2: Study*. Accessed: Dec. 25, 2018. [Online]. Available: https://www.thehindu.com/news/cities/Delhi/thermal-power-plants-leading-to-spike-in-so2-no2-study/article8638600.ece
[39] R. Kumar and A. E. Joseph, "Air pollution concentrations of pm 2.5, pm 10 and no. 2, at ambient and kerbsite and their correlation in metro city–Mumbai," *Environ. Monitor. Assessment*, vol. 119, no. 1-3, pp. 191–199, 2006.
[40] *10 Things You Still Need to Know About Air Pollution*. Accessed: Dec. 28, 2018. [Online]. Available: https://timesofindia.indiatimes.com/10-things-you-still-need-to-know-about-air-pollution/listshow/50541168.cms

APEKSHA AGGARWAL received the B.Tech. degree in computer science and engineering from Uttar Pradesh Technical University, Lucknow, India, and the M.Tech. degree in computer science and engineering from Dr. B. R. Ambedkar National Institute of Technology, Jalandhar, India. She is currently pursuing the Ph.D. degree with IIT Roorkee, Roorkee, India. She has published various peer-reviewed papers in leading international journals, conferences, and book chapters on various topics such as data mining, anomaly detection, data clustering, and frequent itemset mining. She has received several prestigious awards and scholarships, including travel grants from world known organizations such as the Women in Machine Learning.

DURGA TOSHNIWAL received the Ph.D. degree from the IIT Roorkee, India, where she is currently a Professor. She has authored or coauthored more than 150 world known international journals and conferences. She has attended, chaired sessions, and has presented her work in various reputed international conferences in USA, U.K., Australia, and Europe. She has received various awards and honors, including the Best Paper Award at several conferences. Some recent ones are the IBM Faculty Award 2012 and 2008, an Award from UNESCO Chair in Data Privacy 2010, and the very prestigious IBM Shared University Research Award 2009 for her research projects. Her research work has also been featured in DataQuest, the leading IT magazine in India in the Data Quest issue of February 15, 2010, in the article titled "Towards a Greener Planet."