Discovering Fuzzy Periodic-Frequent Patterns in Quantitative Temporal Databases

R. Uday Kiran
NICT, Tokyo, Japan
University of Tokyo, Tokyo, Japan
uday_rage@tkl.iis.u-tokyo.ac.jp

C.Saideep
IIIT-Hyderabad
Hyderabad, Telangana, India
saideep.chennupati@research.iiit.ac.in

Penugonda Ravikumar
RGUKT-AP, Idupuplapaya
Andhra Pradesh, India
raviua138@rguktrkv.ac.in

Koji Zettsu
NICT, Tokyo, Japan
zettsu@nict.gov.jp

Masashi Toyoda
University of Tokyo, Tokyo, Japan
Toyoda@tkl.iis.u-tokyo.ac.jp

Masaru Kitsuregawa
NII, Tokyo, Japan
University of Tokyo, Tokyo, Japan
kitsure@tkl.iis.u-tokyo.ac.jp

P. Krishna Reddy
IIIT-Hyderabad
Hyderabad, Telangana, India
pkreddy@iiit.ac.in

***Abstract*—Periodic-frequent pattern mining is a challenging problem of great importance in many applications. Most previous works focused on finding these patterns in binary temporal databases and did not take into account the quantities of items within the data. This paper proposes a novel model of fuzzy periodic-frequent pattern (FPFP) that may exist in a quantitative temporal database (QTD). Finding FPFPs in QTD is a non-trivial and challenging task due to its huge search space. A novel pruning technique, called *improved maximum scalar cardinality*, has been introduced to effectively reduce the search space and the computational cost of finding the desired itemsets. This technique facilitates the mining of FPFPs in real-world very large databases practicable. An efficient algorithm has also been presented to find all FPFPs in a QTD. Experimental results demonstrate that the proposed algorithm is efficient. We also present a case study in which we apply our model to find useful information in air pollution database.**

***Index Terms*—Data mining, knowledge discovery in databases, fuzzy sets, pattern mining, air pollution**

I. Introduction

Frequent pattern mining (FPM) [1] is an important model in data mining with many real-world applications [2]–[5]. A major obstacle encountered by FPM is as follows: *Since the rationale behind mining the $support$ metric-based frequent patterns is to find all patterns that appear frequently in a database, a huge number of patterns are normally generated and most of which might be found uninteresting depending on application or user requirement. Moreover, the computation cost of finding such huge number of patterns may not be trivial.* When confronted with this problem in real-world applications, researchers have tried to effectively reduce the desired resultant set by employing other interestingness measures, such as closed [6], maximal [7], K-most [8], utility [9], occupancy [10] and periodicity [11]. A class of user interest-based frequent patterns generated using $periodicity$ measure are known as **periodic-frequent patterns**. This paper aims to develop a generalized model of periodic-frequent pattern that may exist in a very large quantitative database.

Tanbeer et al. [11] introduced Periodic-Frequent Pattern Mining (PFPM) [11] to discover all those frequent patterns that are occurring at regular intervals in a transactional database. Since then, the problem of finding periodic-frequent patterns has recieved a great deal of attention [12]–[14]. A classic application of PFPM is market-basket analytics. It analyzes how regularly the itemsets are being purchased by the customers. An example of a periodic-frequent pattern is as follows:

$$\{Bat, Ball\} \quad [support = 5\%, periodicity = 1\ hour] \tag{1}$$

The above pattern says that 5% of the customers have purchased the items 'Bat' and 'Ball,' and the maximum duration between any two consecutive purchases containing both of these items is no more than an hour. This predictive behavior of the customers' purchases may facilitate the user in product recommendation and inventory management. Other real-world applications of periodic-frequent pattern mining includes accident data analytics [15] and body sensor data analytics [16]. Mining periodic-frequent patterns has inspired other data mining tasks such as high-utility periodic pattern mining [17], recurring pattern mining [13] and regular pattern mining [16].

The popular adoption and successful industrial application of PFPM has been hindered by the following obstacle: "*Most studies have aimed at finding periodic-frequent patterns in a binary temporal database. Consequently, they are inadequate to find those interesting patterns that are occurring regularly (or periodically) in a quantitative temporal database.*" To address this issue, we introduce a novel model of **fuzzy periodic-frequent pattern** (FPFP) that may exist in a quantitative temporal database (QTD). Herein we present an important business application that has motivated us to study the problem of finding FPFPs in QTD.

Air pollution is a major cause of cardio-respiratory problems for people living in Japan [18]. To tackle this problem, the Japanese Ministry of Environment has set up the Atmospheric Environmental Regional Observation System (AEROS) constituting of several air pollution measuring sensors (or stations) positioned throughout the Japan. The spatial locations of these sensors is shown in Fig. 1. The data generated by these sensors

978-1-7281-6932-3/20/$31.00 ©2020 IEEE

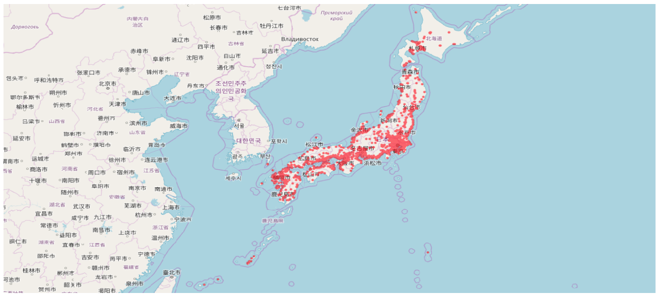
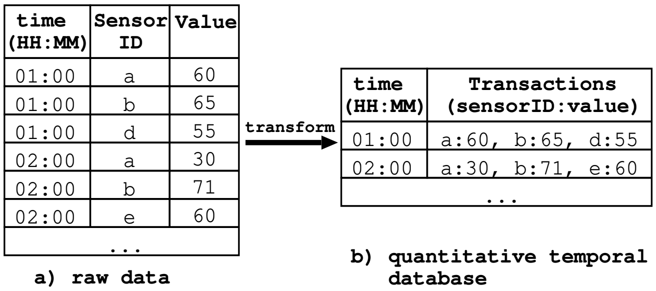


Fig. 1: Spatial visualization of AEROS sensors



time (HH:MM)	Sensor ID	Value
01:00	a	60
01:00	b	65
01:00	d	55
02:00	a	30
02:00	b	71
02:00	e	60
...		

time (HH:MM)	Transactions (sensorID:value)
01:00	a:60, b:65, d:55
02:00	a:30, b:71, e:60
...	

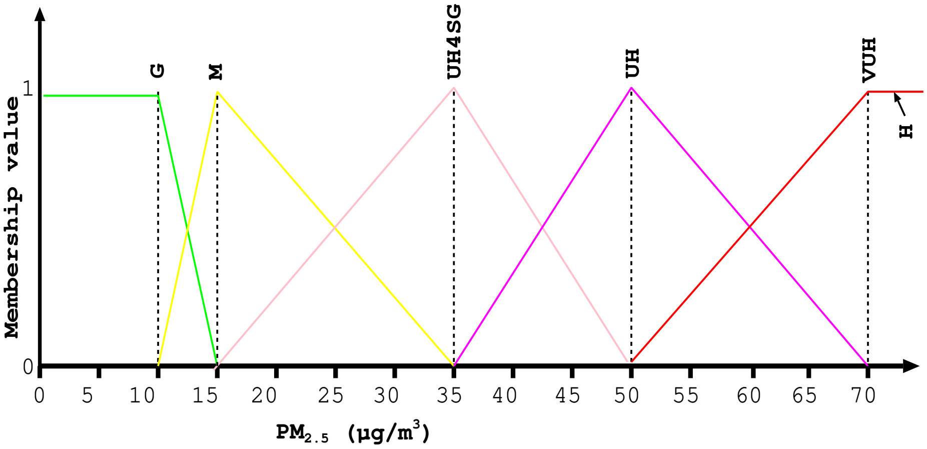
a) raw data

b) quantitative temporal database

Fig. 2: Representing the real-world sensor data as a quantitative temporal database

$PM_{2.5}$ (µg/m^3)	Symbol	Health Descriptor
~10	G	Good
11~15	M	Moderate
16~35	UH4SG	Unhealth for selective groups
36~50	UH	Unhealthy
51~70	VUH	Very Unhealthy
71~	H	Hazardous

(a) $PM_{2.5}$ categories



(b) Representing $PM_{2.5}$ categories as a fuzzy membership function

Fig. 3: Graphical representation of fuzzy membership function for $PM_{2.5}$

for an air pollutant, say $PM_{2.5}$[1], represents a quantitative temporal database (see Fig. 2). Given the $PM_{2.5}$ categories (see Fig. 3(a)) and the corresponding fuzzy membership function (see Fig. 3(b)), a generated fuzzy periodic-frequent pattern, say $\{a.UH4SG, d.UH, g.H\}$ (format: sensorID.Symbol of health descriptor), represents a set of geographical regions where people have been regularly exposed to harmful air pollutants. This information may be found very useful to the environmentalists in taking appropriate steps such as increasing the vegetation, alerting the people to remain indoors at certain times of the day, and proposing new policies to regulate industrial emissions.

Finding FPFPs in a QTD is a non-trivial and challenging task because of the two main reasons:

1) The space of items in a database gives raise to an itemset lattice. This huge itemset lattice (i.e., $2^{|I|} - 1$ itemsets, where $|I|$ denotes the total number of items in a database) represents the search space of finding FPFPs in a QTD. New upper-bound measures have to be invested to effectively reduce the search space and the computational cost of finding the desired FPFPs in a QTD.
2) Several algorithms have been described in the literature to find fuzzy frequent patterns in a quantitative transactional database [19]–[21]. Unfortunately, these algorithms cannot be directly extended to find FPFPs in a QTD. It is because these algorithms do not take into account the temporal occurrence information of a pattern in the data. Consequently, new algorithms that take into account both the quantitative and temporal information of the items within the data needs to be invested to find FPFPs in QTD.

With this motivation, we propose a more flexible model of FPFP that may exist in a QTD. A pattern is said to be a FPFP in QTD if its *support* is no less than the user-specified *minimum support* ($minSup$) and the *periodicity* is no more than the user-specified *maximum periodicity* ($maxPer$). The $minSup$ controls the minimum number of transactions in which an itemset must appear in the database. The $maxPer$ controls the maximum time interval in which a pattern must reappear in the database. A novel pruning technique has been introduced to effectively reduce the search space and the computational cost of finding the desired patterns. An efficient algorithm, called Fuzzy Periodic-Frequent Pattern-Miner (FPFP-Miner), has also been presented to find all desired patterns in a QTD. Experimental results show that the

[1] $PM_{2.5}$ refers to atmospheric particulate matter that have diameter of ≤ 2.5 microns. The unit of measurement is $\mu g/m^3$.

proposed algorithm is efficient. We also present a case study in which we applied the proposed model to identify useful information in air pollution data.

The remainder of this study is organized as follows. In Section 2, we discuss the literature on fuzzy frequent pattern mining and periodic-frequent pattern mining. In Section 3, we introduce the proposed model of fuzzy periodic-frequent patterns. In Section 4, we introduce the proposed algorithm. In Section 5, we present the experimental results. Finally, in Section 6, we provide conclusion of this study and discuss future research directions.

II. Related Work

Frequent pattern mining is an important model in data mining with many real-world applications [22]–[24]. The model involves discovering all patterns in a transactional database that satisfy the user-specified minimum support ($minSup$) constraint. The $minSup$ controls the minimum number of transactions that a pattern must cover within the database. The popular adoption and successful industrial application of this model has been hindered by the following two key limitations:

1) The model implicitly assumes that all items within the data are binary by nature. Consequently, the model fails to discover interesting patterns that are occurring frequently in a quantitative transactional database.
2) The basic model of frequent patterns ignores the temporal occurrence information of a transaction in a database. As a result, the model is inadequate to find those interesting frequent patterns that are occurring periodically (or regularly) in the data.

Chan et al. [25] described a model to find fuzzy frequent patterns (or association rules) that may exist in a quantitative transactional database. An algorithm, called F-APACS, was also presented to find all desired patterns. Hong et al. [26] stated the fuzzy data mining approach to discover fuzzy frequent patterns (FFIs) in a level-wise way. Hong et al. [27] presented an efficient algorithm to merge the same fuzzy sets of the transformed transactions into smaller transformed databases, thus speeding up the computations for level-wisely mining the fuzzy frequent patterns. Jerry et al. [20] recently presented an efficient algorithm, called fuzzy frequent itemset (FFI)-Miner algorithm, to mine fuzzy frequent itemsets efficiently. Although the above approaches consider the quantities of items within the data, they ignore the crucial information pertaining to the temporal occurrence information of the transactions (or patterns) within the data. Consequently, they are inadequate to find user interest-based fuzzy periodic-frequent patterns in a quantitative temporal database.

Tanbeer et al. [11] described a model to find periodic-frequent patterns in a binary temporal database. The model aims to discover all patterns that satisfy the user-specified $minSup$ and *maximum periodicity* ($maxPer$) constraints. The $maxPer$ controls the maximum time interval between any two consecutive occurrences of a pattern in the entire database. Efficient sequential [28]–[30] and parallel [31], [32] algorithms have been described in the literature to find periodic-frequent patterns in very large databases. As the basic model of periodic-frequent pattern implicitly assumes that all items within the data exist in binary form, they are inadequate to find interesting patterns in quantitative temporal database.

Overall, current studies aim to find either fuzzy frequent patterns in a quantitative transactional database or periodic-frequent patterns in a binary temporal database. This papers aims to develop a unified model that tries to find fuzzy periodic-frequent patterns that may exist in a quantitative temporal database. In other words, the proposed model generalizes the current models of fuzzy frequent patterns and periodic-frequent patterns.

III. Proposed Model

Let $I = \{i_1, i_2, \cdots, i_m\}$, $m \geq 1$, be a finite set of m distinct items (or attributions). A quantitative temporal database, QTD, is an ordered collection of transactions and their associated timestamps. Each transaction in this database contains items and their associated quantities. That is, $QTD = \{(1, T_1), (2, T_2), \cdots, (ts, T_{ts})\}$, where $ts \in \mathbb{R}^+$ represents the timestamp and each transaction $T_q \in QTD$, $1 \leq q \leq ts$, is a subset of I, contains several items with its purchase quantities v_{i_q}. A set of items $Y \subseteq I$ is called an itemset (or a pattern). An itemset containing k, $k \geq 1$, number of items is called a k-itemset. An itemset Y is said to be contained in a transaction T_q if $Y \subseteq T_q$.

Example 1. Let $I = \{a, b, c, d, e, f\}$ be the set of items (or sensors recording the volume of an air pollutant, say $PM_{2.5}$). An hypothetical quantitative temporal database generated from the recording of the items in I is shown in Table I. This database contains 12 transactions. Each transaction in this database is associated with a timestamp (ts). In the first transaction, $(1, \{a : 60, b : 65, d : 55\})$, 1 represents the timestamp and $\{a : 60, b : 65, d : 55\}$ represents the transaction containing items and their associated quantities.

TABLE I: Running example: Quantative temporal database

ts	$itemset$
1	$a : 60, b : 65, d : 55$
2	$a : 30, b : 70, e : 60$
3	$a : 55, c : 20$
4	$a : 60, b : 65, d : 55$
5	$a : 55, d : 60, f : 30$
6	$b : 55, c : 40, e : 45$
7	$a : 45, b : 60, c : 45, e : 25$
8	$a : 55, d : 60$
9	$a : 60, b : 65, d : 30$
10	$a : 45, d : 40, f : 40$
11	$a : 60, b : 55, c : 65, d : 55$
12	$b : 45, e : 65$

Definition 1. *Let $\{1, 2, \cdots, h\}$ be the set of fuzzy terms for a membership function μ. The set of linguistic variables that can be drawn from the membership function μ for an item i, denoted as $R_i = \{R_{i1}, R_{i2}, \cdots, R_{ih}\}$, where R_{ik}, $1 \leq k \leq h$ is the fuzzy term mapped to an item i.*

Example 2. The set of fuzzy terms for the quantitative temporal database shown in Table I are: G, M, UH4SG, UH, VUH and H (see Fig. 3(a)). Consequently, the set of fuzzy terms for an item a in Table I, i.e., $R_a = \{a.G,\ a.M,\ a.UH4SG,\ a.UH,\ a.VUH,\ a.H\}$. Same can be stated for the remaining items in the table.

TABLE II: Fuzzy temporal database generated from Table I

ts	$itemset$
1	$a.UH : 0.5, a.VUH : 0.5, b.UH : 0.25, b.VUH : 0.75, d.UH : 0.75, d.VUH : 0.25$
2	$a.M : 0.25, a.UH4G : 0.75, b.VUH : 1, e.UH : 0.5, e.VUH : 0.5$
3	$a.UH : 0.75, a.VUH : 0.25, c.M : 0.75, c.UH4G : 0.25$
4	$a.UH : 0.5, a.VUH : 0.5, b.UH : 0.25, b.VUH : 0.75, d.UH : 0.75, d.VUH : 0.25$
5	$a.UH : 0.75, a.VUH : 0.25, d.UH : 0.5, d.VUH : 0.5, f.M : 0.5, f.UH4G : 0.5$
6	$b.UH : 0.75, b.VUH : 0.25, c.UH4G : 0.66, c.UH : 0.33, e.UH4G : 0.33, e.UH : 0.66$
7	$a.UH4G : 0.33, a.UH : 0.66, b.UH : 0.5, b.VUH : 0.5, c.UH4G : 0.33, c.UH : 66, e.M : 0.5, e.UH4G : 0.5$
8	$a.UH : 0.75, a.VUH : 0.25, d.UH : 0.5, d.VUH : 0.5$
9	$a.UH : 0.5, a.VUH : 0.5, b.UH : 0.25, b.VUH : 0.75, d.M :, d.UH4G :$
10	$a.UH4G : 0.33, a.UH : 0.66, d.UH4G : 0.66, d.UH : 0.33, f.UH4G : 0.66, f.UH : 0.33$
11	$a.UH : 0.5, a.VUH : 0.5, b.UH : 0.75, b.VUH : 0.25, c.UH : 0.25, c.VUH : 0.75, d.UH : 0.75, d.VUH : 0.25$
12	$b.UH4G : 0.33, b.UH : 0.66, e.UH4G : 0.33, e.UH : 0.66$

Definition 2. *Let v_{iq} denote the quantitative value of an item i in the transaction T_q. The fuzzy set, denoted as f_{iq}, is the set of fuzzy terms with their membership degrees (fuzzy values) transformed from the quantitative value v_{iq} of the linguistic variable i by the membership functions μ as:*

$$\begin{aligned} f_{iq} &= \mu_i(v_{iq}) \\ &= \frac{fv_{iq1}}{R_{i1}} + \frac{fv_{iq2}}{R_{i2}} + \cdots + \frac{fv_{iqh}}{R_{ih}} \end{aligned} \quad (2)$$

where, h is the number of fuzzy terms of i transferred by μ, R_{il} is the l-th fuzzy terms of i, fv_{iql} is the membership degree (fuzzy value) of v_{iq} of i in the l-th fuzzy terms R_{il} and $fv_{iql} \in$ $[0, 1]$

Example 3. Consider the item a in Table I. The quantity of a in the first transaction is 60. Thus, $v_{a1} = 60$. Based on the membership function shown in Fig. I, the fuzzy set of a in T_1, i.e., $f_{a1} = \frac{0}{a.G} + \frac{0}{a.M} + \frac{0}{a.UH4SG} + \frac{0.5}{a.UH} + \frac{0.5}{a.VUH} + \frac{0}{a.H} = \frac{0.5}{a.UH} + \frac{0.5}{a.VUH}$. For simplicity, we represent $f_{a1} = \{a.UH : 0.5, a.VUH : 0.5\}$. Similarly, for the items b and d in T_1 , $f_{b1} = \{b.UH : 0.25, b.VUH : 0.75\}$ and $f_{d1} = \{d.UH : 0.75, d.VUH : 0.25\}$. Other transactions in the database can be processed in the similar way. The fuzzy temporal database generated from the Table I is shown in Table II.

Definition 3. *Let QTD' denote the fuzzy temporal database generated from the QTD using the fuzzy membership function μ. The support of the transformed fuzzy terms, denoted $sup(R_{il})$, is the summation of scalar cardinality of the fuzzy values of fuzzy term R_{il}, which can be defined as:*

$$sup(R_{il}) = \sum_{R_{il} \subseteq T_q \wedge T_q \in QTD'} fv_{ilq} \quad (3)$$

Example 4. Table II shows the fuzzy temporal database generated for the quantitative temporal database shown in Table I. The item $d.UH$ appears in the transactions whose timestamps are $1, 4, 5, 8, 10$ and 11. Thus, the $support$ of item $d.UH$, i.e., sup($R_{d.UH}$)=$fv_{d.UH_1} + fv_{d.UH_4} + fv_{d.UH_5} + fv_{d.UH_8} + fv_{d.UH_{10}} + fv_{d.UH_{11}}$=0.75 + 0.75 + 0.5 + 0.5 + 0.33 + 0.75=3.58. Similarly, the item $a.UH$ appears in the transactions whose timestamps are $1, 3, 4, 5, 7, 8, 9, 10$ and 11. Thus, $sup(R_{a.UH})$=$fv_{a.UH_1} + fv_{a.UH_3} + fv_{a.UH_4} + fv_{a.UH_5} + fv_{a.UH_7} + fv_{a.UH_8} + fv_{a.UH_9} + fv_{a.UH_{10}} + fv_{a.UH_{11}}$=0.5 + 0.75 + 0.5 + 0.75 + 0.66 + 0.75 + 0.5 + 0.66 + 0.5=5.58

Definition 4. *The support of fuzzy k-itemsets ($k \geq 2$), denoted as $sup(X)$, is the summation of scalar cardinality of the fuzzy values for X, which can be defined as*

$$sup(X) = \{X \in R_{il} | \sum_{R_{il} \subseteq T_q \wedge T_q \in QTD'} min(fv_{aql}, fv_{bql}) \quad (4)$$

where, $a, b \in X$ and $a \neq b$.

Example 5. The set of fuzzy terms, $\{a.UH, d.UH\}$, is an itemset (or a pattern). This pattern contains 2 items. Therefore, it is a 2-pattern. In Table II, the pattern $\{a.UH, d.UH\}$ occurs in the transactions whose timestamps are $1, 4, 5, 8, 10$ and 11. The $support$ of $\{a.UH, d.UH\}$ in Table II, i.e., $sup(a.UH, d.UH)$=$min(fv_{a.UH_1}, fv_{d.UH_1}) + min(fv_{a.UH_4}, fv_{d.UH_4}) + min(fv_{a.UH_5}, fv_{d.UH_5}) + min(fv_{a.UH_8}, fv_{d.UH_8}) + min(fv_{a.UH_{10}}, fv_{d.UH_{10}}) + min(fv_{a.UH_{11}}, fv_{d.UH_{11}}) = min(0.5, 0.75) + min(0.75, 0.5) + min(0.75, 0.5) + min(0.75, 0.5) + min(0.66, 0.33) + min(0.5, 0.75) = 0.5 + 0.5 + 0.5 + 0.5 + 0.33 + 0.5 = 2.83$

Definition 5. *(**A period of** X) If $X \subseteq T_q$, it is said that X occurs in the transaction T_q. Let ts_q^X denote the timestamp of the transaction T_q containing X. Let ts_i^X and ts_j^X, $ts_{min} \leq i \leq j \leq ts_{max}$, denote two consecutive timestamps at which X has occurred in TDB. A period of X in QTD, denoted as $p_k = ts_j^X - ts_i^X$.*

Example 6. For the simplicity of period computation, the first and the last transactions (say, ts_f and ts_l) in QTD are respectively identified as null (i.e., $ts_f = 0$) and ts_l (i.e., $ts_l = max(ts) + 1$). For instance, the pattern $\{a.UH, d.UH\}$, is observed at timestamps $\{1, 4, 5, 8, 10, 11\}$. Therefore, all periods for the pattern $\{a.UH, d.UH\}$ are: $1(= 1 - ts_f), 3(= 4 - 1), 1(= 5 - 4), 3(= 8 - 5), 2(= 10 - 8), 1(= 11 - 10) and 2(= ts_l - 11)$, where $ts_f = 0$ and $ts_l = 13$.

Definition 6. *($Periodicity$ **of** X.) Let $P^X = \{p_1^X, p_2^X, \cdots, p_k^X\}$, $k = sup(X) + 1$, denote the set of all periods of X in the database. The $periodicity$ of X, denoted as $per(X)$, represents the maximum value among all of its periods. That is, $per(X) = max(p_1^X, p_2^X, \cdots, p_k^X)$.*

TABLE III: FPFP's generated for working example

S.No.	*Pattern*	*Support*	*Periodicity*
1	$\{d.UH\}$	3.58	3
2	$\{d.UH, a.UH\}$	2.83	3
3	$\{b.VUH\}$	4.25	2
4	$\{b.VUH, a.UH\}$	2.25	3
5	$\{a.UH\}$	5.58	2

Example 7. Continuing with the previous example, the set of all periods of $\{a.UH, d.UH\}$, i.e., $P^{\{a.UH,d.UH\}} = \{1, 3, 1, 3, 2, 1, 2\}$. Thus, the $periodicity$ of $\{a.UH, d.UH\}$, i.e., $per(\{a.UH, d.UH\}) = max(1, 3, 1, 3, 2, 1, 2) = 3$.

Definition 7. ***(Fuzzy periodic-frequent pattern** X.**)*** *A pattern X is called a fuzzy periodic-frequent pattern if its $periodicity$ is no greater than the user-specified* maximum periodicity *($maxPer$) and $support$ is no less than the user-specified* minimum support *($minSup$). In other words, X is a fuzzy periodic-frequent pattern if $per(X) \leq maxPer$ and $sup(X) \geq minSup$.*

Example 8. If the user-specified $minSup = 2$ and $maxPer = 3$, then the itemset $\{a.UH, d.UH\}$ is said to be a fuzzy periodic-frequent pattern because $per(\{a.UH, d.UH\}) \leq minPer$ and $sup(\{a.UH, d.UH\}) \geq minSup$. The above pattern provides the useful information that the sensors a and d have frequently observed unhealthy levels of $PM_{2.5}$. This information may be found very useful to the environmentalists in devising appropriate policies to control air pollutants. The complete set of FPFP's generated from Table I are shown in Table III.

Definition 8. ***(Problem definition.)*** *Given the quantitative temporal database (QTD) and the user-specified fuzzy membership function (μ),* minimum support *($minSup$) and* maximum periodicity *($maxPer$), the problem of fuzzy periodic-frequent pattern mining involves discovering all patterns in QTD that have $sup(X) \geq minSup$ and $per(X) \leq maxPer$.*

IV. Proposed Algorithm

The proposed FPFP-Miner employs a fuzzy-list structure to record the fuzzy information of the items in a QTD'. The fuzzy-list structure can be used to efficiently and effectively speed up the computations for directly discovering FPFPs. The phases of the proposed FPFP-Miner algorithm are described below.

A. Fuzzy Periodic frequent 1-patterns (FPFP-1)

To find FPFP-1's, an improved maximum scalar cardinality strategy is adopted, thus making the number of transformed terms used in later processing equal to the number of the original items. This strategy can be used to find the most represented term of each item in the original database.

Definition 9. ***Improved maximum scalar cardinality:*** *For a linguistic variable i, the fuzzy terms R_{il} with the maximum scalar cardinality (support) among the **transformed periodic fuzzy terms** is used to present the linguistic variable (item).*

d.UH		
ts	**if**	**rf**
1	0.75	0.75
4	0.75	0.75
5	0.5	0.75
8	0.5	0.75
10	0.33	0.66
11	0.75	0.5

b.VUH		
ts	**if**	**rf**
1	0.75	0.5
2	1.0	0
4	0.75	0.5
6	0.25	0
7	0.5	0.66
9	0.75	0.5
11	0.25	0.5

a.UH		
ts	**if**	**rf**
1	0.5	0
3	0.75	0
4	0.5	0
5	0.75	0
7	0.66	0
8	0.75	0
9	0.5	0
10	0.66	0
11	0.5	0

Fig. 4: Fuzzy lists of fuzzy-periodic items

After that, the fuzzified quantitative database is then used to find the FPFP-1's. The represented fuzzy terms are considered as the FPFP-1's if the items $support$ is greater than or equal to $minSup$ and $periodicity$ is no more than the $maxPer$. The FPFP-1's of each transformed transaction are sorted in their support-ascending order. If two or more items have same $support$, then those items are sorted in $periodicity$ descending order. This strategy can be used to easily find the fuzzy values between the transformed fuzzy terms based on the designed fuzzy-list structures explained in the next subsection

Definition 10. ***The support-ascending order.*** *For the remaining fuzzy terms with their fuzzy values in a transaction T_q, the fuzzy terms are sorted in their support-ascending order to perform the intersection operation for discovering their support values among the fuzzy k-items ($k \geq 2$)*

Based on the proposed improved maximum scalar cardinality and the support-ascending order strategies, the original databases can be transformed as the fuzzified databases.

B. Periodic Fuzzy list structure

After the original quantitative database is transformed, the FPFP-1's are used to build their own fuzzy-list structures for keeping the fuzzy information. The definitions used in the fuzzy-list structure are respectively given below

Definition 11. *A fuzzy term R_{il} in transaction T_q, and $R_{il} \subseteq T_q$. The set of fuzzy terms after R_{il} in T_q is denoted as $\frac{T_q}{R_{il}}$.*

Definition 12. *The internal fuzzy value of a fuzzy term R_{il} in transaction T_q is denoted as $if(R_{il}, T_q)$.*

Definition 13. *The resting fuzzy value of a fuzzy term R_{il} in transaction T_q is denoted as $rf(R_{il}, T_q)$ by performing the union operation to get the maximum fuzzy value of all the fuzzy terms as the upper-bound value in T_q/R_{il} in T_q, which is defined as:*

$$rf(R_{il}, T_q) = max(if(z, T_q) | z \in T_q/R_{il}) \quad (5)$$

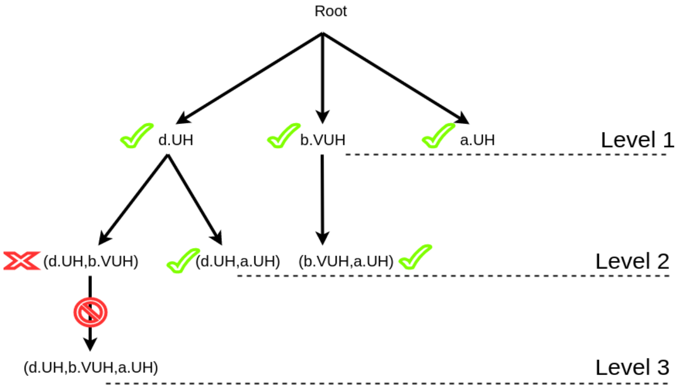


Fig. 5: Enumeration tree for the QTD

In the constructed fuzzy-list structure, each element consists of three attributes as:

- **Timestamp** (ts), which indicates temporal occurence of T_q containing R_{il}.
- **Internal fuzzy value** (if), which indicates the fuzzy value of R_{il} in T_q.
- **Resting fuzzy value** (rf), which indicates the maximum fuzzy value of the resting fuzzy terms after R_{il} in T_q.

The initial fuzzy-list structures of the fuzzy terms in L1 are first constructed. Since the support-ascending order of the fuzzy terms in L_1 are ($d.UH < b.VUH < a.UH$), the results are shown in Fig. 4. The construction algorithm of fuzzy-list structure is shown in Algorithm 1.

Algorithm 1 Construct
Input :$P_X.FL$, $P_Y.FL$ the fuzzy-lists of P_X and P_Y.
Output :$P_{XY}.FL$ the fuzzy-list of P_{XY}

```
initialise $P_{XY}.FL$ to Null
for each $E_Y \in P_X$ do
    if $\exists E_Y \in P_Y.FL$ and $E_X.tid = E_Y.tid$ then
        $E_{XY}.tid \leftarrow E_X.tid$
        $E_{XY}.if \leftarrow min(E_X.if, E_Y.if)$
        $E_{XY}.rf \leftarrow E_Y.rf$
        $E_{XY} \leftarrow \langle E_{XY}.tid, E_{XY}.if, E_{XY}.rf \rangle$
        append $E_{XY}$ to $P_{XY}.FL$
return $P_{XY}$
```

C. Search space of FPFP-miner

Based on the designed fuzzy-list structure, the search space of the proposed FPFP-Miner algorithm can be represented as an enumeration tree according to the developed support-ascending order strategy. In this example, the search space of the enumeration tree is shown in Fig. 5. Since the complete search space of the enumeration tree is very huge for discovering all fuzzy periodic frequent patterns, it is necessary to reduce the search space but still can completely find the fuzzy periodic frequent patterns

Strategy 1. For an pattern X, if its $SUM(X.if)$ is no less than the $minSup$ and $periodicity(X)$ less than $maxPer$, it is considered as a Fuzzy periodic frequent pattern. Also, if $min(SUM(X.if), SUM(X.rf))$ of X is no less than the minimum support count, the supersets of X are required to be generated and determined

Strategy 2. If the summation of the resting fuzzy values of the itemset X is no larger than the minimum support count, any extensions of X will not be a periodic fuzzy frequent itemsets and can be directly ignored to avoid the construction phase of the fuzzy-list structures of the extensions of X.

The approach for this is clearly stated in Algorithm 2

Algorithm 2 *FPFP-Miner*
Input:FL_{prefix}, prefix, FPFP-1, $maxPer$ and $minSup$.
Output : Fuzzy periodic patterns

```
while FPFP-1 $\neq$ Null do
    $FPFP_{suffix}$ = FPFP-1.pop()
    $FL_{(prefix\|suffix)}$=$construct(FL_{prefix}, FPFP_{suffix})$
    if $(sum(FL_{(prefix\|suffix)}.if) \geq minSup)$ AND $period((FL_{prefix\|suffix}) \leq maxPer)$ then
        Generate $(prefix \| suffix)$ as a FPFP.
        if $sum(FL_{(prefix\|suffix)}.rf) \geq minSup$ then
            $FPFP\text{-}Miner(FL_{(prefix\|suffix)}, (prefix \| suffix), FPFP\text{-}1)$
```

V. Experimental results

Since there exists no algorithm to find FPFPs in a QTD, we only evaluate the proposed FPFP-Miner algorithm using the real-world database. We also demonstrate the usefulness of the proposed patterns with a real-world application.

The FPFP-Miner algorithm has been written in python language and executed on a machine with Ubuntu 18.0 on a 2.66 GHz Intel i5 machine with 8 GB of RAM.

Two real-world databases, **Air Pollution** and **Mushroom**, were used for evaluating algorithms. **Air pollution** database is generated based on the recordings of sensors between 1-April-2018 to 30-April-2018 containing 1600 items (or stations) with 720 transactions. It is a high dimensional dense database with minimum, average and maximum transaction lengths equal to 11, 460 and 971, respectively and the membership function for air pollution data is same as that in Fig. 3(b). **Mushroom** database is a popular real-world dense database containing 8,124 transactions and 119 distinct items. It is a dense database with minimum, average and maximum transaction lengths equal to 23, 23 and 23, respectively. Each item in the Mushroom database has been assigned with a random number $n \in (1, 11)$. The membership function used for the Mushroom database is shown in Fig. ?.

Fig. 6(a) and 7(a) respectively show the number of FPFPs generated in Air pollution and Mushroom databases at different $minSup$ and $maxPer$ values. The following observations can be drawn from these figures: (i) the $maxPer$ constraint has positive effect on the generation of FPFPs. It is because the increase in $maxPer$ facilitates the patterns with high inter-arrival times to be considered as periodic in the data. (ii) the $minSup$ constraint has negative effect on the generation of

TABLE IV: Few interesting patterns generated in air pollution database

S.No.	Patterns (or station ids)	color
1	{4135.M, 4307.M, 4275.M, 3930.M, 4256.M}	Yellow
2	{4055.M,4517.M,4574.M,45606.M}	Yellow
3	{1264.G}	Green
5	{5218.G}	Green
6	{4462.UH4SG}	Red
7	{4209.UH4SG}	Red
8	{4733.UH4SG}	Red
9	{3656.UH4SG}	Red

FPFPs. It is because more patterns fail to satisfy the increased $minSup$ value.

Fig. 6(b) and 7(b) respectively show the memory consumed by the FPFP-Miner in Air pollution and Mushroom databases at different $minSup$ and $maxPer$ values. The following observations can be drawn from these figures: (i) increase in $maxPer$ has resulted in the increased memory requirements of FPFP-Miner algorithm. It is because the algorithm has to spend additional resources to accommodate the increase the fuzzy periodic-frequent patterns. (ii) increase in $minSup$ has resulted in the decrease of memory requirements of FPFP-Miner algorithm. It is because the algorithm has to spend fewer resources to fewer FPFPs from the QTD.

Fig. 6(c) and 7(c) respectively show the runtime requirments of FPFP-Miner in Air Pollution and Mushroom databases at different $minSup$ and $maxPer$ values. As the runtime requirements of FPFP-Miner depend on the number of patterns being generated, similar observations as that of the generated patterns can be drawn from these figures.

A. A case study: identifying highly polluted PM2.5 regions in Japan

Table IV shows the FPFPs generated in the air pollution database at $minSup = 100$ and $maxPer = 19\ hrs$. The spatial location of all these stations in the entire Japan are shown in Fig. 8. It can be observed that high levels of PM2.5 has been regularly observed in Tokyo and its surrounding prefectures. Especially, air pollution levels in some areas in Tokyo are unhealthy for selective groups. This information can be found very useful in devising policies to control pollution at bay areas.

VI. Conclusions and Future Work

In this paper, we have introduced a flexible model of fuzzy periodic-frequent pattern that may exist in a quantitative temporal database. A novel pruning technique has been introduced to effectively reduce the search space and the computational cost of finding the desired patterns effectively. An efficient fuzzy-list based set enumeration tree approach has also been presented to find all desired patterns in QTD. Experimental results demonstrate that the proposed algorithm is efficient. Finally, we have also demonstrated the usefulness of the proposed model with a real-world case study on air pollution data.

In this paper, we have studied the problem of finding FPFPs by taking into account positive quantities of items within the data. As a part of future work, we would like to investigate finding FPFPs in a QTD using both positive and negative weights for the items. Additionally, we would like to investigate disk-based and parallel algorithms to find FPFPs.

References

[1] R. Agrawal, R. Srikant *et al.*, "Fast algorithms for mining association rules," in *Proc. 20th int. conf. very large data bases*, vol. 1215, 1994, pp. 487–499.

[2] C. C. Aggarwal, *Applications of Frequent Pattern Mining*. Springer International Publishing, 2014, pp. 443–467.

[3] P. Fournier-Viger, J. C.-W. Lin, R. U. Kiran, and Y. S. Koh, "A survey of sequential pattern mining," *Data Science and Pattern Recognition*, vol. 1, no. 1, pp. 54–77, 2017.

[4] F. THABTAH, "A review of associative classification mining," *The Knowledge Engineering Review*, vol. 22, no. 1, pp. 37–65, Mar. 2007.

[5] N. Abdelhamid and F. Thabtah, "Associative classification approaches: Review and comparison," *Journal of Information and Knowledge Management*, vol. 13, no. 03, pp. 1–30, 2014.

[6] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal, "Discovering frequent closed itemsets for association rules," in *Proceedings of the 7th International Conference on Database Theory*, 1999, pp. 398–416.

[7] K. Gouda and M. J. Zaki, "Efficiently mining maximal frequent itemsets," in *Proceedings of the 2001 IEEE International Conference on Data Mining*, 2001, pp. 163–170.

[8] A. Salam and M. S. H. Khayal, "Mining top-k frequent patterns without minimum support threshold," *Knowl. Inf. Syst.*, vol. 30, no. 1, pp. 57–86, Jan. 2012.

[9] R. U. Kiran, T. Y. Reddy, P. Fournier-Viger, M. Toyoda, P. K. Reddy, and M. Kitsuregawa, "Efficiently finding high utility-frequent itemsets using cutoff and suffix utility," in *Advances in Knowledge Discovery and Data Mining - 23rd Pacific-Asia Conference, Part II*, 2019, pp. 191–203.

[10] L. Tang, L. Zhang, P. Luo, and M. Wang, "Incorporating occupancy into frequent pattern mining for high quality pattern recommendation," in *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, 2012, pp. 75–84.

[11] S. K. Tanbeer, C. F. Ahmed, B.-S. Jeong, and Y.-K. Lee, "Discovering periodic-frequent patterns in transactional databases," in *Advances in Knowledge Discovery and Data Mining*, 2009, pp. 242–253.

[12] R. U. Kiran and M. Kitsuregawa, "Finding periodic patterns in big data," in *Big Data Analytics - 4th International Conference*, 2015, pp. 121–133.

[13] R. U. Kiran, H. Shang, M. Toyoda, and M. Kitsuregawa, "Discovering recurring patterns in time series," in *Proceedings of the 18th International Conference on Extending Database Technology*, 2015, pp. 97–108.

[14] P. Fournier-Viger, Z. Li, J. C. Lin, R. U. Kiran, and H. Fujita, "Efficient algorithms to identify periodic patterns in multiple sequences," *Inf. Sci.*, vol. 489, pp. 205–226, 2019.

[15] J. N. Venkatesh, R. U. Kiran, P. K. Reddy, and M. Kitsuregawa, "Discovering periodic-correlated patterns in temporal databases," *T. Large-Scale Data- and Knowledge-Centered Systems*, vol. 38, pp. 146–172, 2018.

[16] S. K. Tanbeer, M. M. Hassan, A. Almogren, M. Zuair, and B. Jeong, "Scalable regular pattern mining in evolving body sensor data," *Future Generation Comp. Syst.*, vol. 75, pp. 172–186, 2017.

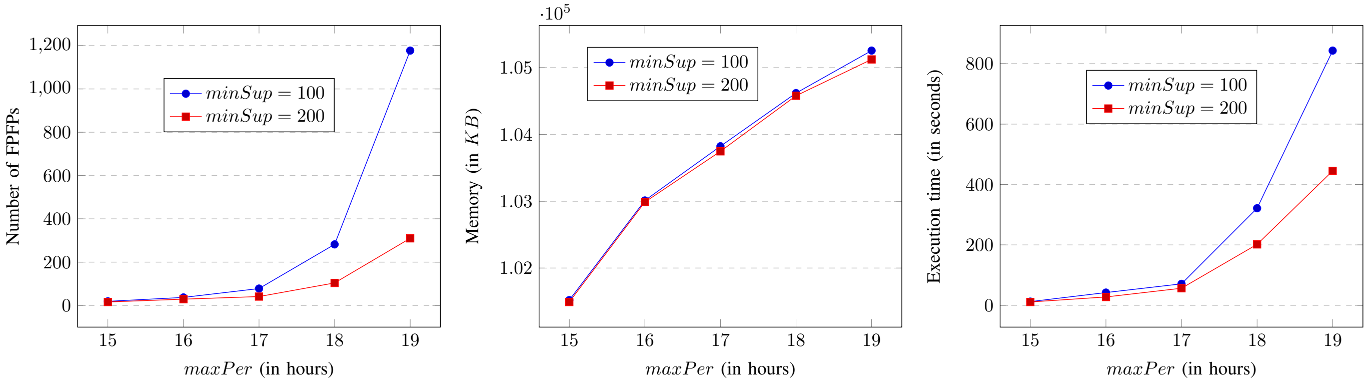
[17] D. Dinh, B. Le, P. Fournier-Viger, and V. Huynh, "An efficient algorithm for mining periodic high-utility sequential patterns," *Appl. Intell.*, vol. 48, no. 12, pp. 4694–4714, 2018.

[18] M. Dao and K. Zettsu, "Discovering environmental impacts on public health using heterogeneous big sensory data," in *2015 IEEE International Congress on Big Data, New York City, NY, USA, June 27 - July 2, 2015*, 2015, pp. 741–744.

[19] C. Lin, T. Hong, and W. Lu, "Linguistic data mining with fuzzy fp-trees," in *Expert Systems with Applications*, 2010, pp. 4560–4567.

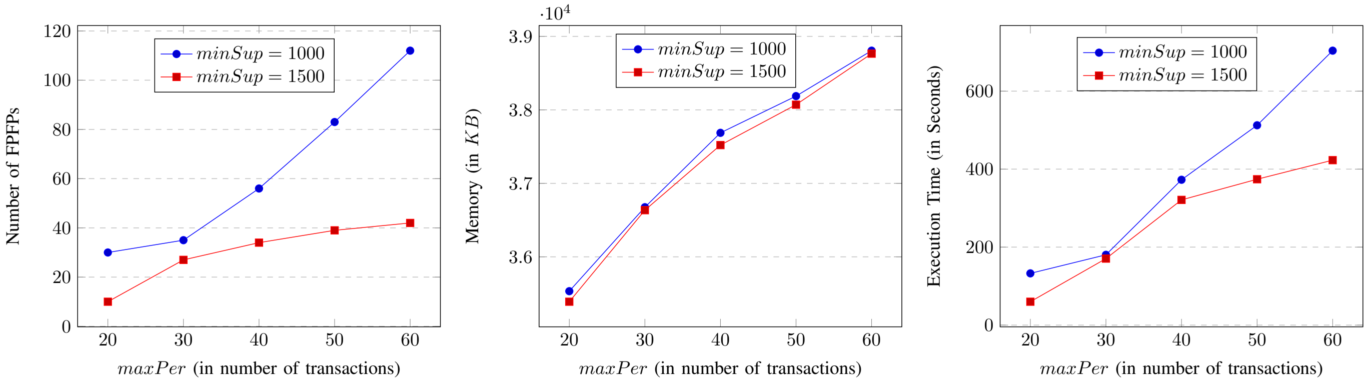
[20] C.-W. Lin, T. Li, P. Fournier Viger, and T.-P. Hong, "A fast algorithm for mining fuzzy frequent itemsets," vol. 29, 10 2015, pp. 2373–2379.

[21] C. Lin, T. Hong, and W. Lu, "An efficient tree-based fuzzy data mining approach," in *International Journal of Fuzzy System*, 2010, pp. 150–157.



(a) Pattern Count (b) Memory (c) Execution Time

Fig. 6: Evaluation of FPFP-miner on Air Pollution data



(a) Pattern Count (b) Memory (c) Execution Time

Fig. 7: Evaluation of FPFP-miner on Mushroom data

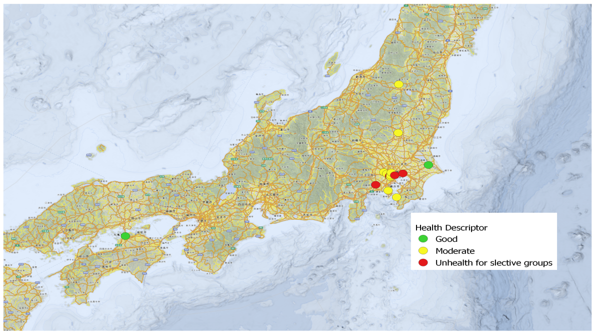


Fig. 8: Some of the interesting patterns generated in air pollution database

[22] J. Han, J. Pei, Y. Yin, and R. Mao, “Mining frequent patterns without candidate generation: A frequent-pattern tree approach,” *Data Min. Knowl. Discov.*, vol. 8, no. 1, pp. 53–87, Jan. 2004.

[23] T. Calders, N. Dexters, and B. Goethals, “Mining frequent itemsets in a stream,” in *ICDM*, 2007, pp. 83–92.

[24] S. Srivastava, R. U. Kiran, and P. K. Reddy, “Discovering diverse-frequent patterns in transactional databases,” in *COMAD*, 2011, pp. 69–78.

[25] K. C. C. Chan and W. H. Au, “Mining fuzzy association rules,” in *Proceedings of the International Conference on Information and Knowledge Management*, 1997, pp. 209–215.

[26] T. P. Hong, C. S. Kuo, and S. Chi, “Mining association rules from quantitative data,” in *Intelligent Data Analysis*, 1999, pp. 363–376.

[27] T. P. Hong, G. C. Lan, Y. H. Lin, and S. T. Pan, “An effective gradual data-reduction strategy for fuzzy itemset mining,” in *International Journal of Fuzzy Systems*, 2013, pp. 170–181.

[28] R. U. Kiran, J. N. Venkatesh, P. Fournier-Viger, M. Toyoda, P. K. Reddy, and M. Kitsuregawa, “Discovering periodic patterns in non-uniform temporal databases,” in *Advances in Knowledge Discovery and Data Mining - 21st Pacific-Asia Conference, Part II*, 2017, pp. 604–617.

[29] A. Anirudh, R. U. Kirany, P. K. Reddy, and M. Kitsuregaway, “Memory efficient mining of periodic-frequent patterns in transactional databases,” in *2016 IEEE Symposium Series on Computational Intelligence*, 2016, pp. 1–8.

[30] F. Rasheed, “Efficient periodic pattern mining in time series & sequence databases,” Ph.D. dissertation, Calgary, Alta., Canada, Canada, 2011, aAINR75499.

[31] H. Li, Y. Wang, D. Zhang, M. Zhang, and E. Chang, “Pfp: Parallel fp-growth for query recommendation,” 01 2008, pp. 107–114.

[32] A. Alampally, U. Rage, P. Krishna Reddy, M. Toyoda, and M. Kitsuregawa, “An efficient map-reduce framework to mine periodic frequent patterns,” 08 2017, pp. 120–129.